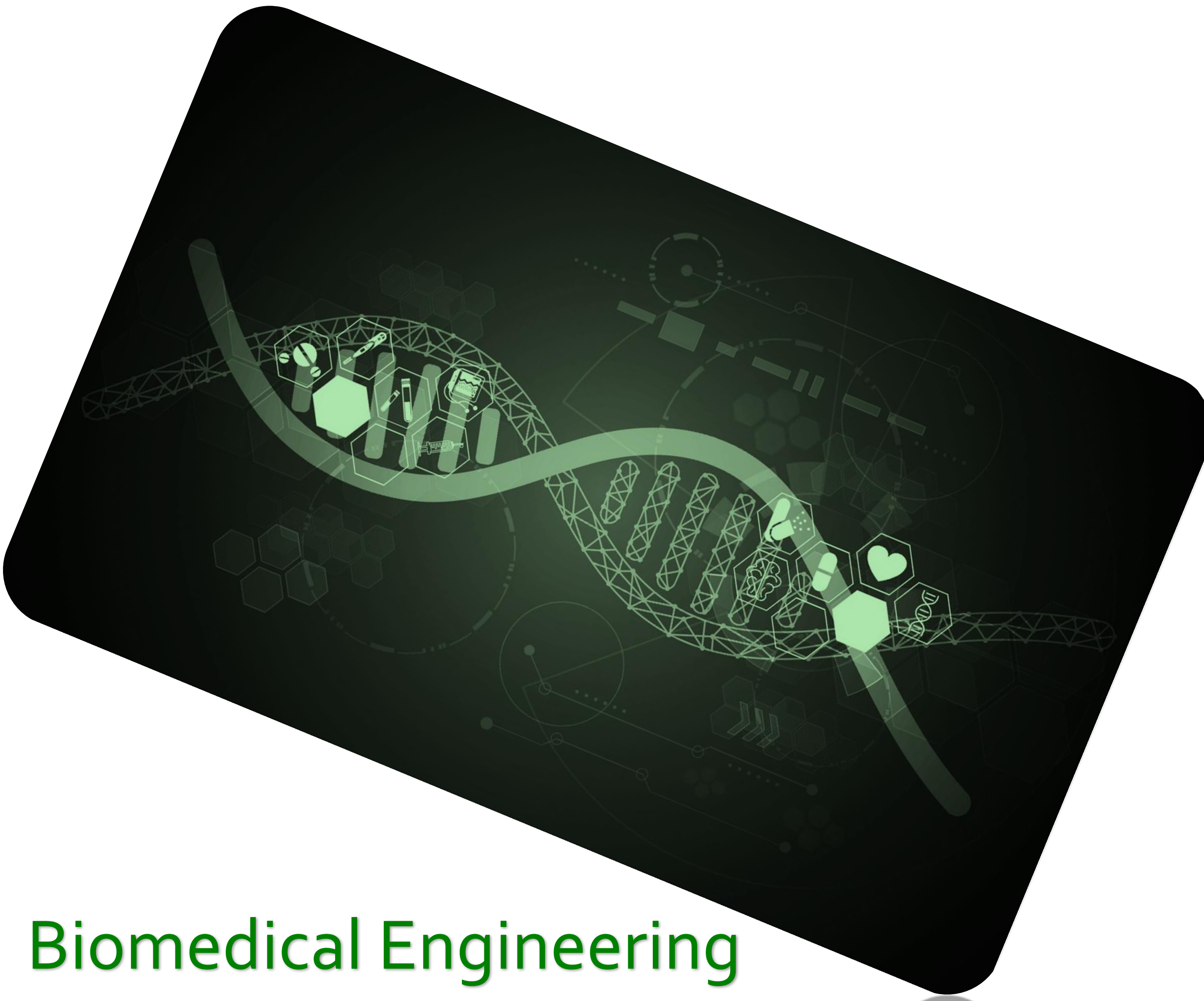


# Biomedical Engineering Unpacked

Solution Development Pack

## Planning Stage

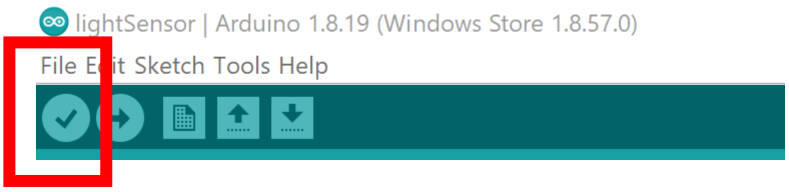
## C++ Rules:

- End your coding instructions with **;**
- Variables have limited scope **{ }**
- Every bracket needs its partner
- Comments for code are represented with **//**

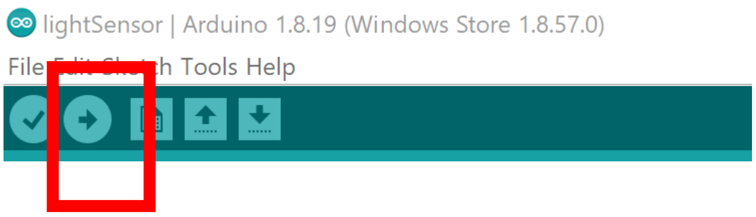
## Create, Test, Improve

We have grouped the creation, testing, and improvement phases together since these three phases feed into each other with this project. To determine if you have 'created the code' correctly, we have to test the code and then discover where we are able to make improvements.

To test if the computer understands your code – press the verify button.


lightSensor | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help

To upload your code to the microcontroller – press the upload button.


lightSensor | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help

## Exercise 1: Making a Pattern of Lights

Using the provided starter code blinkingLights.io on the laptop and the provided reference sheet – try to achieve the following:

1) Get the first NeoPixel to glow red and experiment with its brightness
   a. Brightness = 255
   b. Brightness = 125
   c. Brightness = 0
2) Flash all the odd NeoPixels Red
3) Flash all the 10 NeoPixels a different Colour
4) Change the NeoPixel colour once the program has started

It is important to remember that in C++, we begin counting at 0 and there are 10 NeoPixels.

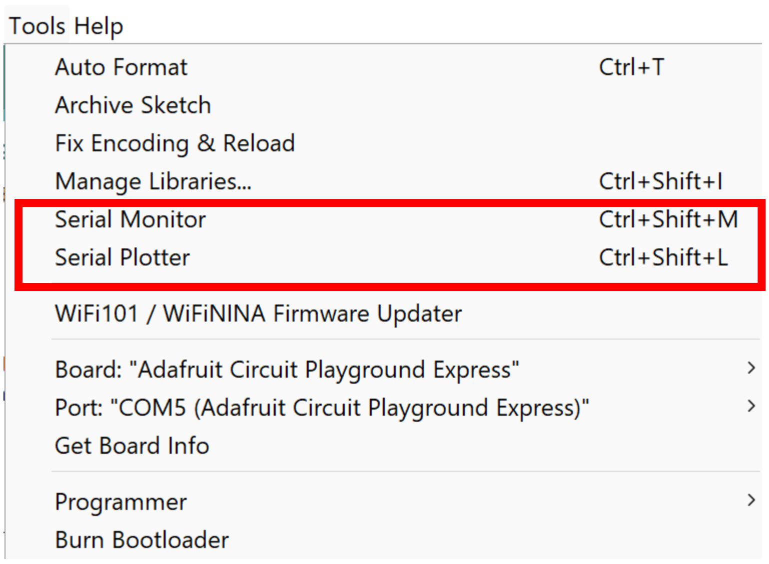
Additionally, bear in mind that the NeoPixel's colour is defined using the RGB spectrum where R represents red, G represents green, and B represents blue. To get any colour, we merely manipulate the amount of these three values. Below is an example of the RGB components for different colours.

| Color | HTML / CSS Name | Decimal Code (R,G,B) |
|---|---|---|
| ⬛ | Black | (0,0,0) |
| ⬜ | White | (255,255,255) |
| 🟥 | Red | (255,0,0) |
| 🟩 | Lime | (0,255,0) |
| 🟦 | Blue | (0,0,255) |
| 🟨 | Yellow | (255,255,0) |
| 🟦 | Cyan / Aqua | (0,255,255) |
| 🟪 | Magenta / Fuchsia | (255,0,255) |
| ⬜ | Silver | (192,192,192) |
| ⬛ | Gray | (128,128,128) |
| 🟫 | Maroon | (128,0,0) |

## Exercise 2: Understanding the Light Sensor and The Serial Plotter

Using the provided starter code lightSensor.io and the reference sheet, try and see if you can measure changes in light intensity. Once you have completed the code, plot the value of this light intensity change to the screen.

To access these values, go to the Tool bar and select Tools and then either the Serial Monitor (gives the numerical values) or the Serial Plotter (plots a graph with the values).


Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Adafruit Circuit Playground Express"
Port: "COM5 (Adafruit Circuit Playground Express)"
Get Board Info
Programmer
Burn Bootloader

To manipulate the value, hold your finger on the light sensor and then remove it. Additionally, try shining your cell phone torch on the light sensor. What happens?

# Exercise 3: Putting It All Together

Now that you understand how to manipulate the NeoPixels, the light sensor and the serial plotter, we can begin to generate our pulse waveforms.

Using the starter code pulseWaves.io and the reference sheet, generate a pulse wave plot. Start off with green light and your index finger. Also use the NeoPixel closest to the light sensor.

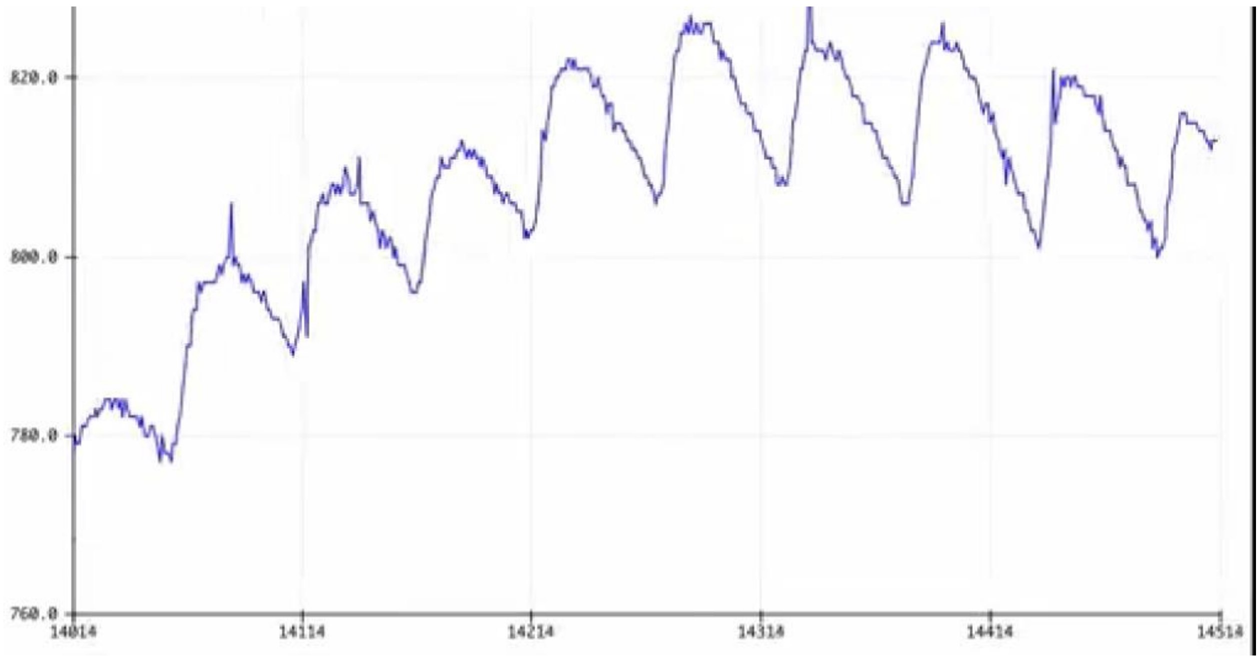
It is important to remember that the quality of the wave is affected by the pressure of your finger on the light sensor. Try a firm pressure and reduce until the waveform matches what is expected. Additionally, the waveform is manipulated by movement so try to keep still.

Also remember that this is reflective PPG, meaning we measure not the light that the blood vessel absorbs but rather the returned light reflected to the light detector. How might this affect the readings? Is there anything you may have to do to get the correct value?

**Hint:** If you start off with 100ml of water and drop a blueberry inside and 80ml of water remains, how much water represents the blueberry's weight?

The maximum reading an Arduino can read is 1024.

**The waveform should look like:**



Try the following:

1) Different fingers on the light sensor
2) Try your ear lobe on the light sensor (get your partner to help with this)
3) Try different colours of light for the NeoPixels

# Reference Sheet

The reference sheet serves to provide a understanding of possible functions to use during this activity.

# Circuit Playground Functions

## General Circuit Playground Functions

Initialize the Circuit Playground Express Board in the setup() function

```
CircuitPlayground.begin();
```

## Controlling the NeoPixels

Neopixels are the lights on the Adafruit Circuit Playground Express board. There are 10 lights in total.

### 1) Set and Change the Displayed Colour

The colour of each NeoPixel is defined by the mixture of three colours: Red, Green and Blue. This is referred to as the RGB color model.

```
CircuitPlayground.setPixelColor(i, iRed, iGreen, iBlue);
```

`i` is the number of the NeoPixel

`iRed` is the magnitude of the red color, 0 ≤ `iRed` ≤ 255

`iGreen` is the magnitude of the red color, 0 ≤ `iGreen` ≤ 255

`iBlue` is the magnitude of the red color, 0 ≤ `iBlue` ≤ 255

You can use constants for the values of `i`, `iRed`, `iGreen` and `iBlue`. For example

```
CircuitPlayground.setPixelColor(3, 200, 0, 200);
```

### 2) Turn off all NeoPixels

```
CircuitPlayground.clearPixels();
```

### 3) Set and Change Brightness

This function is typically called once at the beginning of your code, in the **setup()** function, to set the overall brightness for all NeoPixels. It can not be used to change the brightness of a single NeoPixel.

**CircuitPlayground.setBrightness(i);**

**i** is the value you want to set the brightness to and must be between 0-255.

### 4) Showing all the NeoPixels

**CircuitPlayground.strip.show();**

## Reading in from the Light Sensor

The Circuit Playground Express has an on-board light sensor. The Circuit Playground software library handles the low-level details.

The reading of the light sensor is obtained with the statement

**value = CircuitPlayground.lightSensor();**

where **value** is an **int**. The reading is an 8-bit value, i.e. an integer in the range between 0 and 1023, where 0 is completely dark and 1023 is the maximum brightness the sensor can record. Typical ambient light levels are about 300 on this scale.

## Arduino Functions

## delay(ms)

Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in a second.)

**ms**: the number of milliseconds to pause.

## Serial.println(value)

Prints data to the serial port as human-readable text followed  a newline.

**value**: the value to print out to the screen.

**HINT: the starter code will have variables defined before the void startup()  section to help you in the rest of the code.**

## Reference List

Arduino.cc. 2022. *delay() - Arduino Reference*. [online] Available at: <https://www.arduino.cc/reference/en/language/functions/time/delay/> [Accessed 17 July 2022].

Arduino.cc. 2022. *Serial.println() - Arduino Reference*. [online] Available at: <https://www.arduino.cc/reference/en/language/functions/communication/serial/println/> [Accessed 17 July 2022].

Barela, A., 2022. *Circuit Playground Bike Light*. [online] Adafruit Learning System. Available at: <https://learn.adafruit.com/circuit-playground-bike-light?view=all> [Accessed 17 July 2022].

Rapid Tables. n.d. *RGB Color Codes Chart*. [online] Available at: <https://www.rapidtables.com/web/color/RGB_Color.html> [Accessed 17 July 2022].

Recktenwald, G., 2021. *On-board the CPX*. [online] Invention Bootcamp 2021. Available at: <https://web.cecs.pdx.edu/~gerry/class/IB2021/programming/on-board/> [Accessed 17 July 2022].

Rice University, 2020. *Visualize Your Heartbeat - Maker Challenge*. [online] Teach Engineering. Available at: <https://www.teachengineering.org/makerchallenges/view/rice3-2349-heartbeat-microcontroller-led-senor-design-challenge> [Accessed 17 July 2022].