**Lab #3a: Introduction to the Arduino**

No-write up for this component of the lab. We will continue next week with experiments, and you will have writeups for these subsequent components.

**Purpose:**

In this lab you will create your first program for the Arduino and learn how to interface with it. You should be using your Arduino Student Kit for this work. If you don't yet have that, please acquire one before next week, available at the College Bookstore and online at:
https://store.arduino.cc/usa/student-kit

**Online resources:**

1. http://engineering.nyu.edu/gk12/amps-cbri/pdf/ArduinoBooks/Arduino%20Programming%20Notebook.pdf
2. https://www.arduino.cc/en/Tutorial/HomePage

**Tasks:**

1. Install the Arduino IDE (interactive development environment):
   https://www.arduino.cc/en/main/software
   *Do not use the online web editor, things work much better with the desktop software.

2. Connect your Arduino computer to your laptop/desktop computer using the USB cable.

3. From within the Arduino IDE, you will need to make sure that you have a proper communications connection. This requires two things:
   a. You have the proper communication port selected. To verify this, go to top menu bar and select Tools > Port. You should see something like "COM Port 4 (Arduino/Genuino Uno)" selected as the default. If this is not the case, then select the COM port associated with the Arduino/Genuino Uno tag.
   b. You have the proper board selected. There are many different kinds of Arduino boards and each kind requires a slightly different way of communicating. Make sure that your board is set to the "Arduino/Genuino Uno" device.

4. Write code to cause the on-board LED to flash. You can find example code here:
   https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink

5. Change the code to cause the LED to blink in a Long-Short-Long pattern.

6. Change the code again to define the time delays using variables. If you aren't sure how to do this (this is likely the case) then do a google search for something like "Arduino code defining constants" or use some of the resources listed above to figure out how to do this. Or, you can post a question to Piazza to get help from the class.

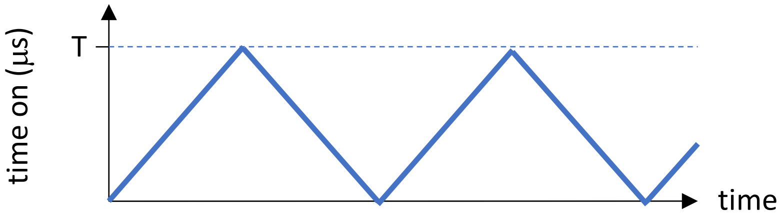
7. **The challenge:** Create a variable dimmer for the LED. Recall last week when we saw this work using the PWM (pulse width modulation) output to the red LED on the breadboard… we want to do something similar here, but the output that is connected to the built-in LED can't do PWM, so we need to write our own function for that.

To make this happen you need to do a few things:

a. We need to decrease the time scale to get a smoother function. To do this we can replace the delay() function with the delayMicroseconds() function.

b. We need to change the duty cycle at each iteration of the loop. To do this, we can think about adjusting the **timeon** variable between 0 and some maximum value (the period, let's call this **T**). In your code you will then want to set the LED to HIGH for this **timeon,** and set it the LOW for **T-timeon**. The sum of the time that it is HIGH plus the time that it is LOW is then always equal to **T**.

c. You want the duty cycle to have the following shape:



d. One way to think about this curve is that when the duty cycle curve has a positive slope it is increasing by the same amount each time. When it gets to the max value, you want the slope to reverse, so you want the change in the duty cycle to change sign. When it gets down to 0% duty cycle you want the change to flip sign again.

e. If you need help figuring this out then you can search for things like "Arduino PWM LED" or "Arduino LED fade" or you can post a question to the lab forum in Piazza.

Hints for the last part:

- You will need to include three new elements in the program header (the header is the code that appears before the start of the **setup** function).
  - One of these describes the total time period (this should be around 1000 or so) and should be declared a constant since you won't ever change it.
  - The next one should be the variable for the amount of time that your LED will be on, perhaps with a nice descriptive name like **Ton**. Since this value changes we do not want to declare it using const. Same with the next variable.
  - The last one is the amount by which you will increase the on-time with each loop. You might call this something like **dt**, reminding us of the time step used in calculus.
- At each loop you want to turn on the LED, delay for some time, turn off the LED, and delay for some amount of time (you need to write a formula to do this part). Then you need to do the following:
  - Change the amount of time that the LED will be on for the next cycle.
  - You want to check to make sure that the on-time has not exceeded the max time, or gone below the minimum time (zero). To do this you need to use two "if statements" that allow you to do some conditional math. For example, if the value of the on-time is greater than the max, then you want to change the sign of dt.