ECE 241 **Lab 1** **Spring 2023**
10 points

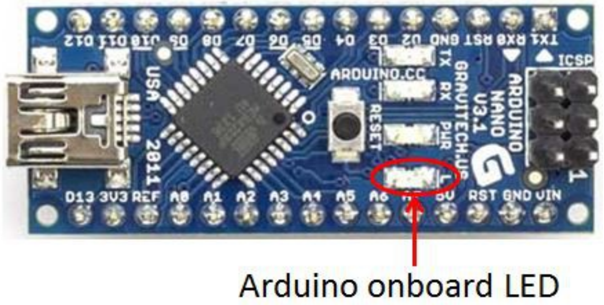
Lab 1: Arduino and their Integrated Development Environment (IDE)

Objective: Familiarize the student with the Arduino system, the Arduino IDE, and the proper way to measure time on the Arduino.

Description of Lab 1:

In the past, this type of programming assignment was known as a “Hello World” assignment, since the objective was to simply print “Hello World” on the display. As simple as this task appears, one learns a lot from such an assignment, such as how to enter a program, compile it, and execute it on the computer.

In the case of an embedded computer, we don't always have a screen on which to display a message. Thus a common first program is to simply blink a light on the board. These programs are commonly called Blinky light programs. In this assignment we will be working on the “proper” way to blink the Light Emitting Diode (LED) on the board. There is an LED connected to pin 13 and we will work to flash it as close to 1 second and as efficiently as possible.



Arduino onboard LED

Figure 1-1. Arduino Nano with onboard LED.

Now one of the most important things is to ***not use the delay function*** provided by the Arduino IDE. The problem with the delay function is that it will tie up the processor, doing nothing but waiting for a certain amount of time to pass. Rather, we will want to continue looping until the requested time has passed. This continued looping can be achieved by using the built-in timer function millis().

Relevant Web Page: https://learn.adafruit.com/multi-tasking-the-arduino-part-1/using-millis-for-timing

The function millis() returns the number of milliseconds that have passed since the system was powered up. Thus, if we were to take a reading of this function at the start of your program and then test the current value of millis() each time you go through your loop, you will be able to tell when a second has passed, and still keep looping until that time.

The structure of the program would look something like this

Psuedo Code (English description)

```
declare an unsigned long named LedTimer

SetUp:
    Set LED pin to Output using pinMode()
    Record value of millis() into the variable LedTimer,

Loop:
    if millis() - LedTimer is greater than or equal to 1000
        if LED pin is high using digitalRead()
            Set LED pin low using digitalWrite()

        else
            Set LED pin high using digitalWrite()

        Update LedTimer by adding 1000 to it.
```

C code .

```
unsigned long LedTimer;   // =>

// =>
void setup() {
  pinMode( 13, OUTPUT ); // =>
  LedTimer = millis(); // =>
 } // =>

// =>
void loop() {
  if( millis() - LedTimer >= 1000 ) { //  =>
      if( digitalRead(13) == HIGH ) { //  =>
         digitalWrite( 13, LOW );     //  =>
      }
      else { //  =>
          digitalWrite( 13, HIGH );
      } //  =>
      LedTimer += 1000;  // =>

   } //  =>

} //  =>
```

Functions Needed for Lab: pinMode() , digitalWrite(), digitialRead(), and millis()

Documentation for all for these functions can be found at the Arduino website

http://www.arduino.cc/en/Reference/HomePage .

Additional programming instruction is available, examples of which are given here.

https://www.youtube.com/watch?v=nXvy5900m3M

https://www.youtube.com/watch?v=-CpG3oATGIs

https://www.youtube.com/watch?v=rk2fK2IIiiQ

Prelab: (As in prior to coming to the lab) The code shown above has no real commenting, however at multiple places the comment `// =>` can be seen. You should make a file that uses the code (above on the right), filling in the comments, replacing the `=>` with a statement explaining the operation done by that line of code. This code needs to be placed in a document file. Do not upload the .ino file built by the Arduino IDE, since Canvas does not know how to read an ino file. Then upload the document to the assignments prelab link.

HINT: The code on the previous page has been placed on the Canvas page for this course under "Files>Lab Write Ups>Lab 01"

It should be noted that the Arduino IDE is free online at http://www.arduino.cc/en/Main/Software and is available on most of the Engineering computers. Thus you can write and test much of the code for the lab, prior to coming into the lab.

In the Lab:

1) Enter in the program as uploaded as your prelab, upload it to the Arduino, and verify that the LED changes every second. The functioning program is to be demonstrated to your lab instructor.

2) Adapt the program to toggle a second pin (pin 12) every 3 seconds. This can be done by establishing a second timer (LedTimer3), which detects when 3 seconds has passed. The functioning program is to be demonstrated to your lab instructor.

Since only one LED is available on the Arduino, the second pin will have to be monitored via the logic probe. Its usage will be demonstrated in class and in the lab.

3) The report should document your program, but also explain the objective of the program and the results. An example report can be found on Canvas under "Files>Lab Write Ups".