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

Objective: Familiarization with the Arduino and its software environment.

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.

Well in the case of an embedded computer, we don't always have a screen on which to display a message, so instead of displaying a message on a screen, we commonly start by simply blinking a light on the board.

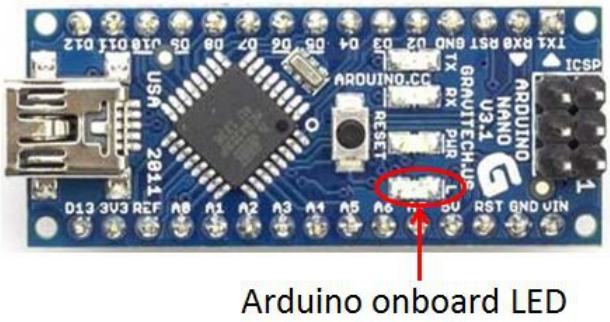

Figure 1-1. Arduino Nano with onboard LED.

These programs are commonly called Blinky light programs. In this assignment we will be working on the "proper" way to blink the LED on the board connected to pin 13 as close to 1 second and as efficiently as possible.

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 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().

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;  // =>
  } //  =>
} //  =>
```

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

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

Lab Assignment:

Prelab: 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 c 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 that is then uploaded to the assignments prelab assignment.

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 ECE computers. Thus you will

Lab 1:

1) Enter in the program shown above, 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.

   Note: Only one LED is available on the board, so a second pin will need to be attached to an LED to be placed on the proto-board. A demonstration of how the proto-board can be used and how to hook up the second LED will be done in class and by the lab TA.

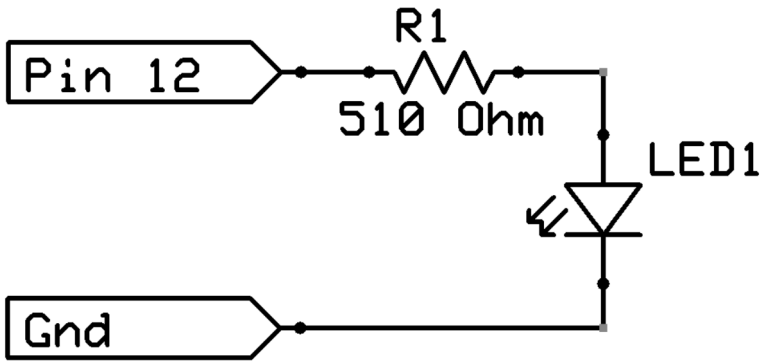

Figure 1-2. Schematic of Added LED.

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.