IAT 884
Week 3
Intro to Micro-controllers: Arduino Outputs

Preparation:

1. Visit the Arduino website and familiarize yourself with the Arduino Microcontroller's functionality and hardware specs.

2. Download and install the Arduino software on your laptop: http://www.arduino.cc/en/Main/Software

Installation Instructions:

Windows: Follow the instructions here: http://www.arduino.cc/en/Guide/Windows
**Please note that you will not be installing the default Windows driver.*

Macintosh: Follow the instructions here: http://arduino.cc/en/Guide/MacOSX

Linux: http://www.arduino.cc/playground/Learning/Linux

Required Reading:
In *Programming Interactivity*: Chapter 4 (p. 91-128)
*For this week focus on pages 91-100, 102-107, 115-122, 126-128.

In Physical Computing read:
Digital Output: p.87 - 89
Analog Output: p.102 - 104

Suggested reading and resources:
Arduino software download page: (http://www.arduino.cc/en/Main/Software)
Arduino programming language reference: http://www.arduino.cc/en/Reference/HomePage
Arduino Hardware: http://www.arduino.cc/en/Guide/Board
Intro to Arduino Course from TodBots:
http://todbot.com/blog/wp-content/uploads/2006/10/arduino_spooky_projects_class1.pdf

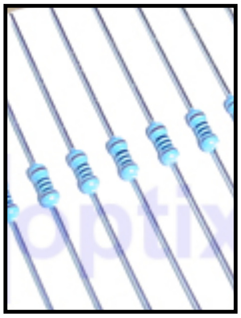
IAT 884
Lab Week 3
Intro to Micro-controllers: Arduino Outputs

In Class Exercise

For this workshop you will be developing small projects using the Arduino microcontroller and the accompanying programming environment.

Materials:

- Arduino Board
- USB Cable
- Breadboard
- 2 x LEDs
- Wire
- 2 x 1k Ohm Resistors (Brown, Black and Red striped)



Tasks:

1. Digital Out: Make 2 LEDs blink so that when one is on the other is off.

2. PWM Out: Make 2 LEDs fluctuate in brightness. The two LEDs should be in sync so that when one LED is lit, the other is completely dark.

3. Optional Challenge: Use keystrokes to dim/brighten an LED (Uses Serial Communication).

Serial Data Reference:

To start serial communication you must open the serial port at a specific baud rate:

```
Serial.begin(9600)
```

To send data (A.K.A. write to the serial port) you use the command:

`Serial.write(data);` or `Serial.Writeln(data);`

To write an ASCII character use the BYTE format. To write a value between 0-255 use DEC (which is the default.)

`Serial.write(data, BYTE);` or `Serial.write(data, DEC);`

To read data from the serial port that has been sent from an application to the Arduino you use:

```
Serial.read();
```

This will return either an integer, the first byte of serial data available, or a -1 if there is no serial data to read.

The following code will write the number 0 -255 in succession to the serial port.

```
int currentValue = 0;    // variable to hold the analog value

void setup() {
        // open the serial port at 9600 bps:
        Serial.begin(9600);
      }

void loop() {
       for(i=0; i<256; i++){
             Serial.println(currentValue, DEC); (Prints number between 0-255)
             Serial.println(currentValue, BYTE); (Prints ASCII character)
             delay(500); //wait ½ sec before taking next serial reading
      }
      currentValue = 0;
}
```

Notes on sending Serial Data:

From: http://itp.nyu.edu/physcomp/Labs/Serial