CSE P567 - Winter 2010 Name______________________________

Lab 6/Homework 6 – Getting Started with the Arduino Board

You should complete the first 2 parts in Lab (by 9:30PM), and the rest as a homework assignment, which you will demo in the next Lab. There will be a small (10%) penalty for late turnin.

Overview

In this lab you will start using your Arduino board with the Arduino IDE and programming language, which is essentially C with some extensions to support embedded systems programming.

Part 1 – First Circuit

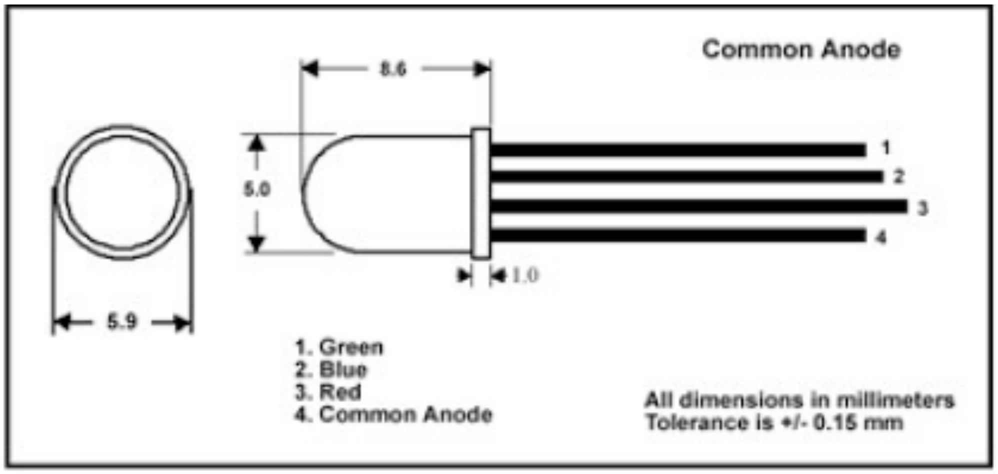
The Arduino home page http://arduino.cc/en/Guide/HomePage has instructions on how to download and use the Arduino IDE. (There is a ton of information on the Arduino Web site that you should be take advantage of.) Follow these instructions to load the IDE on your home machine. The IDE is already installed on the Lab computers. To use the Lab machines, you will need to install the serial drivers when you login – follow the instructions on the Lab Web page.

The last step of the startup instructions is to compile, download and execute the first program, Blink, which simply blinks the light on the Arduino board. Once you have this program running, modify it so that it blinks alternately slow (1 per sec) and fast (4 per sec).

Simple Blinking Light Demo ___________________________TA____________Date

Part 2 – Tri-Color LEDs

Install a tri-color LED on your proto-board. Here is a figure showing this LED:



The datasheet for this tri-color LED is linked to the Lab Web page. If you look at that, you will see that the specs are different for the three colors and thus we have to wire them up differently. The graphs show how brightness varies with current – let's assume that the LEDs will be more or less equally bright with the same current and we'll pick a value of 7.5mA. To get this current, we will place a resistor in series with the LED. We all know that V = IR and we need to solve for R to get the right current. However, there is a voltage drop across the LED so the voltage drop across the resistor is 5v – Vdrop, and the equation for the resistor is:

R = (5v – Vled)/7.5mA

Vled is about 3.5v for the blue and green LEDs and 2v for the red LED which gives us values of 200ohms for the blue and green LEDs and 400ohms for the red LED. Of course, our assumption doesn't really hold, so if you want a true white when all three LEDs are on, you may need to adjust these resistances. (360ohms for the red LED seems to work pretty well.)

Connect the Common Anode to the +5v supply on your protoboard, and connect the Red, Blue and Green pins to the Arduino digital pins 7, 8, and 9 through 200, 100, and 100 ohm resistors respectively. Do this wiring with the board disconnected from power. We will have wire and wire cutters/strippers available for you to use. Feel free to cut yourself some extra wires to take home with you. If you have any question on how to do this, you can take a look at our circuit board.

You will be using digital output pins 7, 8, and 9 to drive these LEDs. Note that the common anode is connected to +5v, which means that current will flow through the LED when the pin is at 0v (LOW). Keep this in mind when writing your program.

Write a program that cycles through the 3 colors: red, green, and then blue, with each on for 1 sec.

Tri-Color LED Demo ________________________________TA________________Date

Part 3 – Program-Driven PWM Control of Tri-Color LEDs

Write a program that uses program-driven pulse-width modulation to cycle through the three colors, but this time each color ramps from off to full brightness to off as illustrated in the following graph. Each color should be given 1 sec. to ramp up and then back down again. A pulse-width modulated signal is essentially a clock that is fast enough that the eye cannot see the pulses. The intensity of the light depends on the "duty cycle" of the clock – the frequency is constant. The duty cycle is the percentage of the clock period that the clock is high and ranges from 0% (no high pulse) to 50% (normal clock signal) to 100% (always high). Your program should generate a clock signal with a frequency of about 10KHz, and vary the duty cycle to ramp the brightness up and down. Use the delayMicroseconds() function or micros() function to do this. You must use the digitalWrite() function to implement this program.

Program-Driven PWM Demo ________________________TA______________Date

Part 4 – Analog PWM Control of Tri-Color LEDs

We will now look at using the analog output pins to drive these LEDs. Some of the digital pins can be used in analog output mode, which really means PWM mode using PWM hardware built into the ATmega pins.

First re-connect the tri-color LED to three analog output pins. Now, re-write your program to use the analogWrite() function to drive these pins with a 10-bit value. The pins will convert this value automagically into a PWM signal, which is a whole lot easier than using a programmed method.

Analog PWM Demo ________________________TA______________Date

Part 5 – Using the Serial Monitor to Control LEDs

There is a Serial library (actually a class) that allows you to communicate with the PC via the USB after your program starts running. Take a look at the program on the next page to see a good example of using Serial. Use the Serial Monitor built into the IDE to enter characters and display the results.

Now write a program to control your tri-color LED via the serial interface. Use a 2-character command where the first character indicates which light to program (0-2 say) and the second character indicates the intensity, from 0-9 say. Your program should echo a long version of the command as a confirmation. (You may embellish your program if you like of course.)

Serial Monitor Control Demo ______________________TA______________Date

```
/*
 * Serial Read Blink
 * -----------------
 * Turns on and off a light emitting diode(LED) connected to digital
 * pin 13. The LED will blink the number of times given by a
 * single-digit ASCII number read from the serial port.
 *
 * Created 18 October 2006
 * copyleft 2006 Tod E. Kurt <tod@todbot.com>
 * http://todbot.com/
 *
 * based on "serial_read_advanced" example
 */
 int ledPin = 13;   // select the pin for the LED
 int val = 0;       // variable to store the data from the serial port
 void setup() {
   pinMode(ledPin,OUTPUT);    // declare the LED's pin as output
   Serial.begin(9600);        // connect to the serial port
 }
 void loop () {
   val = Serial.read();      // read the serial port
   // if the stored value is a single digit, blink the LED that number
   if (val > '0' && val <= '9' ) {
     val = val - '0';          // convert from character to number
     for(int i=0; i<val; i++) {
       Serial.println("blink!");
       digitalWrite(ledPin,HIGH);
       delay(150);
       digitalWrite(ledPin, LOW);
       delay(150);
     }
     //Serial.println();
   }
 }
```