

OPTI 380B Intermediate Optics Laboratory

Lab 9

Microcontroller / Data Acquisition Project I

Objectives: This lab is an introduction to the world of microcontrollers. In particular, we will explore the world of digital input-output "I/O" using an Arduino microcontroller. This tiny computer, originally developed in Italy (2004/2005) to aid non-engineers (i.e. students of art and design) in controlling hardware, is now one of the most popular microcontrollers in the world. Its popularity is due in part to its ease of use, ease of programming, and the fact that its hardware and software are totally open-source. As a result, a variety of companies are selling ".duino" microcontrollers. ("As of 2016, 17 versions of the Arduino hardware had been commercially produced." Wikipedia). The particular version that we will be using (the "UNO Rev.3") is an "official" Arduino and not a clone.

In this lab we will:

- Learn how to program the Arduino, using basic elements of the C++ language.
- Use the pre-written program 'Blink.ino' to flash an LED.
- Modify the 'Blink.ino' program to flash an LED at various frequencies and duty cycles.
- Learn to control the digital I/O pins:
Write and run a program to control the digital logic level of the output pins. Coupled with a simple LED circuit, your program will input a decimal number and visually display its binary equivalent.
- Learn to use the built-in analog-to-digital converter (by reading a voltage at one of the analog input pins):
Write a program and build a simple detector circuit to use the Arduino as a controller for a simple LED night light. The program will turn on an LED when the room lights are turned off.
- Learn how to write information to the 16-character, 2-line LCD display.

Reading Assignment:

- Two books are strongly suggested as being the best (i.e. easiest to read, most useful) sources of information on programming and using the Arduino:

"Programming Arduino: Getting Started with Sketches"
(hereafter referred to as PA)

"Exploring Arduino: Tools and Techniques for Engineering Wizardry"
(hereafter referred to as EA)

Both books are available in the lab as reference books.

Please DO NOT remove them from the lab!

- The official source of information for the Arduino is at: www.arduino.cc
- The supplier of the additional hardware boards we will use is a company called Adafruit: www.adafruit.com Their website is a great source of information for the Data Logger, Motor Controller, and LCD boards ("shields") that we will be using.

Two important points before we begin:

▶ When wiring up a circuit on the breadboard, make power connections to the Arduino last (after double-checking the connections in your circuit). Most likely, the Arduino power pins are short-circuit protected.....but still.....

To turn the power to the Arduino board OFF, simply unplug the USB cable from the main computer.

▶ Do NOT connect an "output" pin directly to ground, or +5V. This will, most likely, destroy the CPU.

USEFUL FACTS:

- Programs written for the Arduino are called "sketches".
- The Arduino has 32KB of memory to store a sketch.
- Circuit boards (i.e. hardware) that plug onto the Arduino are called "shields".
 - Examples include the Data Logging shield, the Motor Controller shield, and the LCD shield that we will be using (all purchased from adafruit.com).
- The Arduino is programmed using a separate computer that is running the IDE ("Integrated Development Environment")—this is a program available free from the Arduino website: www.arduino.cc
- The Arduino has:
 - 6 analog input pins (labeled A0-A5). HOWEVER, pins A4 and A5 are used for the I2C serial bus, which we will use to send data to/from the 3 shields. As a result, in practice, we only have 4 analog input pins available for sampling analog voltages.
 - a built-in (10-bit) analog-to-digital converter (ADC). As a result, analog voltages are sampled and represented by a digital value ranging from 0-1023. The reference voltage that the ADC uses is nominally set (internally) to 5V. To change this, connect a different DC voltage (between 0 and 5 V) to the AREF pin. The maximum digital output of 1023 "counts" corresponds to the reference voltage.
 - 14 digital I/O pins (labeled 0-13). HOWEVER, pins 0 (RX) and 1 (TX) are reserved for serial communications to the main computer, through the USB bus.
 - has 6 "analog output" pins that can supply a DC voltage between 0-5 volts. In fact, these are digital output pins 3, 5, 6, 9, 10, and 11 each labeled with a '~'. They supply a Pulse Width Modulated (PWM) signal at roughly 500Hz, that varies in amplitude (0-5V) and in duty cycle. The RMS equivalent of this PWM signal is effectively a DC voltage ranging from 0-5 V.
- Each of the digital I/O pins can supply a maximum current of 40 mA.
DO NOT EXCEED THIS!!!

THE LABORATORY EXERCISE

A. LAYOUT OF THE ARDUINO MICROCONTROLLER BOARD

Refer to the picture below to familiarize yourself with the Arduino circuit board:



Using the actual Arduino circuit board, and both the PA and EA reference books, answer the following questions:

- [1] What is the identification of the computer chip (CPU) on our boards?
- [2] What is the clock frequency of the crystal?
- [3] What is the tiny surface-mount chip by pin 13? (HINT: 'Blink.ino')
- [4] Where do you make power connections from the Arduino to circuits that you build on the breadboard?
- [5] What is the 'safe' range of input voltages that you can supply to the DC barrel connector, to power the Arduino (instead of using the USB connector)?

B. DIGITAL OUTPUT (“I/O”) – LED FLASHER

- Load the program “Blink.ino” (found in the IDE: File/Examples/01.Basics/Blink)
- Run the program.

- [6] What is the frequency of the flash? (obtained from the code itself)
- [7] What is the duty cycle? (obtained from the code itself)

- Modify the program to vary the frequency (maintain a 50% duty cycle).
- [8] What is the frequency above which you can no longer see the 'flashing'?

C. DIGITAL OUTPUT – BINARY NUMBER

- Wire up the LED bar and the SIP resistor package so that the 8 "right-most" LED's are connected to digital I/O pins 0-7.

- Run the program that you wrote in pre-lab that converts a decimal number to its binary equivalent. Configure your program so that the appropriate LED's light up.

- Run the program. Debug your program until it works properly!!

- [9] For the following 5 decimal numbers, what is the pattern of bits that you observe as the binary equivalent?

	(MSB).....(LSB)
0	= _ _ _ _ _ _ _ _
6	= _ _ _ _ _ _ _ _
17	= _ _ _ _ _ _ _ _
141	= _ _ _ _ _ _ _ _
255	= _ _ _ _ _ _ _ _

D. LED NIGHT LIGHT

- Wire up simple resistor voltage divider circuit (using a photoconductor and a series resistor). Connect the output (the junction of the photoconductor and the resistor) to analog input pin A5.

- Write a program that turns on an LED when the room lights are turned off. (Use the "analogRead(A5)" command to read the voltage at the output of the resistor divider.

E. LCD DISPLAY

- Learn how to write text to the LCD display (see PA ch. 9 pgs. 125-130). Run the program "Write_to_LCD.ino"modify it to write the following:
I LOVE ARDUINO! to the 2-line LCD display ☺
OPTI 380B RULES!

(If time permits, create a custom character!)