cs281: Introduction to Computer Systems

Lab01 – HW Intro

Overview

The objective of this lab is to introduce you to the environment that we will be using as part of the hardware lab sequence in this course. In particular, you will learn about the *breadboard* prototyping stations and about the *Arduino*, a system for supporting embedded processor control. We will start by exploring the features of the breadboard and then will integrate use of the Arduino with the breadboard.

Each team must turn in one lab report. The report must be typeset, and PDF is the preferred format. For this first lab, your team should, in complete and coherent sentences, answer any questions asked below. Be sure that your report is clear about what question is being answered. Lab reports are due by the start of Wednesday lecture.

Breadboard Basics

1. Plug the machine in and turn it on. Peruse the board. There are a number of things we will not be using. Most importantly, the only power supply we need is +5 volts. NEVER hook a wire to any of the other voltages. Make sure all appropriate switches are set to TTL or +5. Follow from the +5 Volt Peg in the upper right of the breadboard to the main breadboard proper. The breadboard is structured so certain holes are wired together underneath the breadboard, but others are isolated from each other. Your first task is to understand what is wired (underneath, or by wires above) to where.

2. Find the set of onboard *Logic Indicators*. There are eight of them and you can use them to determine HIGH or LOW voltage from a location on the breadboard by connecting a wire from the location of interest to one of the eight Logic Indicators. Start with a location where you believe +5 volts is wired to, and test it. Also test a location where you believe GND is wired. Were you right?

3. Continue tracking down where +5 and Ground are available throughout the board. Write down your findings in your lab response. Also note (in a generalization based on your experimentation) which holes are neither HIGH nor LOW. Notice the groove down the middle of each board section. You need to figure out how the little holes on either side of the groove are wired. Devise a series of experiments to do this. Summarize your experiments and conclusions.

4. A *circuit* is a complete path from +5V to GND, with intervening devices and gates. Our first circuit will wire up and light an LED, and we are going to replicate two such circuits. Get two LEDs and two 330 Ohm resistors from your arduino kit and wire them on the breadboard in the following sequence by using independent rows in the breadboard. Note that an LED has a + and a - lead, where the longer wire is + and the shorter is -. On the other hand, a resistor does not have a “direction”, and so either end can be used for + or -. Implement the following circuit sequence:

   (a) A wire from +5V to a row (call it A) on the breadboard.

   (b) The resistor with one lead in row A and the other lead in another row, call it row B.

   (c) +LED in row B, and -LED in another row, call it row C.

   (d) A wire from row C to a GND connection on the breadboard.

Does your LED light? If not, then verify the sequence again, otherwise create a separate circuit with the other LED and resistor.

5. Remove the wire from +5V to row A for each of your LED circuits. On the first circuit, wire from a TTL connection associated with the *Function Generator*, run a wire to row A. Manipulate the controls of the Function Generator and describe in your lab response how the controls work. Try and set the controls so that the Function Generator is operating at exactly 10 Hz. What is the time equivalent units for 1 Hz? Can you set the controls to get 50 Hz? What is the time equivalent?

6. In the next step, we are going to introduce our first *logic gate*, a simple negation. Get a 7404 chip (and a 7408 chip for a later step) from your instructor. The 7404 provides six 1-input NOT gates on a single chip. Consider the two rows of pins on the chip. We need all 14 pins to operate independently of one another. From your conclusion earlier about what holes are independent of each other, how must the chip be inserted onto the breadboard? Put the chip in place with the "notch" oriented toward the top of the board.

7. Each chip needs +5V (VCC) and GND wired to it in order to operate. Using the spec sheet for the 7404, wire +5V and ground to the appropriate rows for the chip pins. Now add a new wire from row A of the first LED circuit (coming from the function generator) to the chip pin for the input to one of the NOT gates; then wire the pin for the output of that NOT gate to row A of your second LED circuit. What do you expect to see? Is your circuit working properly? If not, use the Logic Indicators to test the sequence of inputs and outputs until you figure out the problem.

8. Next put the 7408 chip onto the breadboard. This chip provides 4 independent AND gates, each with two inputs and one output. First wire VCC and GND to the chip, using the spec sheet to guide you. Then wire the output pin for one of the AND gates to the row A of your first LED circuit (unwiring from the Function Generator). Wire the first input pin of the AND gate to the first switch, S_1, at the bottom-left of the breadboard, and the second input of the AND gate to the second switch, S_2. By hand, toggle through all four combinations of HIGH and LOW for the two input switches. Does the lighting of the LED correspond to the truth table of the AND operation? If it does not, then debug your circuit.

9. Now rewire the *second* input to the AND so that its source is the Function Generator, and experiment with toggling the other input switch S_1. Describe the behavior of the circuit.

Arduino Control

One of the uses of an embedded controller, like the Arduino, is that we can specify a behavior for controlling a circuit, and can then have the circuit operate independently (autonomously) of any manual input. So in this part of the lab, we will develop a fairly straightforward equivalent way of driving the circuit you have already implemented on the breadboard.

On your team's laptop, download the Arduino IDE from `http://arduino.cc/en/Main/Software`. The app runs without any other dependencies, so you can choose to either install on your system or just run from wherever you have downloaded.

If you have not already done so, get an Arduino controller kit from your instructor, and then follow these instructions:

1. Log into the laptop and plug the USB cable from the laptop to the socket on the Arduino. If the system presents a dialog not recognizing the USB device, you can simply close the dialog.

2. Start the Arduino IDE.

3. Wire pin 13 on the Arduino to second input pin on AND gate, replacing the input from the Function Generator.

4. Wire +5 from the breadboard to +5 on the Arduino.

5. Wire GND from the breadboard to GND on the Arduino. Although there are 3 pins labeled GND, we typically use the one next to +5V.

6. In the Arduino Sketch IDE, type in the following C++ code:

```
const int P = 13;
const int A = 1000;
const int B = 1000;

void setup() {
  pinMode(P, OUTPUT);
}

void loop() {
  digitalWrite(P, HIGH);
  delay(A);
  digitalWrite(P, LOW);
  delay(B);
}
```

7. To compile and verify your C++ code by clicking the *Verify* button (with "Checkmark" icon) at the top-left of the interface. If compilation is not successful, fix any syntax errors indicated.

8. On the Arduino board, toward the center, there are two mini-LEDs marked Tx/Rx. Observer these LEDs while you click the *Upload* button (the right arrow next to the Verify button). Describe what happens with the TxRx and with your circuit once the upload is complete.

9. Experiment with your sketch program by changing the values of A and B. Infer the units of the values for A and B. Can you change the value of P? If you change the value of P in the software, what corresponding change is required in the hardware setup?

10. Unplug the USB tether between the laptop and the Arduino. What happens? What does this mean as far as *where* the C++ program you wrote is actually running?