**Hardware Lab 0: Using a Prototype Board and Checking Logic Circuits using a Voltmeter**

**Prerequisites:** Before beginning this laboratory experiment you must be able to:

* Describe the following concepts: voltage, current, ground, open circuit, and closed circuit.
* Recognize the schematic symbol for an AND gate.
* Know how to connect logic gates.
* Communicate with the Arduino board using the Firmata Test application. Refer to the instructional videos for Week 2 on how to download and install the Arduino IDE, upload the firmware, download and test Firmata.

**Equipment:** Personal computer, Arduino Uno (or equivalent) with Firmata firmware loaded, USB cable, breadboard, breadboarding wire bundle.

**Integrated Circuits:** You will need the following IC to complete this lab:

* 7408 (Quad 2-input AND gate)

**Objective:** The objective of this laboratory exercise is to familiarize you with the operation of a typical prototyping board and a voltmeter.

**Outcomes:** When you have completed the tasks in this experiment you will be able to:

* Insert an IC in the proper location in a breadboard.
* Wire the IC to ground and the +5 Volt (+5V) power supply.
* Connect pins of an IC to realize a combinational logic circuit.
* Use a voltmeter to determine a signal’s voltage levels.

**Introduction**

In this laboratory exercise you will familiarize yourself with a rapid hardware prototyping platform, consisting of a Breadboard and a microcontroller, which acts as a digital stimulation and measurement interface and also provides the capability of measuring voltage levels. This will enable you to design, build, test and debug hardware digital logic circuits that are covered in the course.

For this tutorial lab you will be building a three-input logic AND gate using a digital logic chip that only has two-input AND gates. This chip will be placed on the breadboard, connected to power and ground, its inputs wired to digital logic outputs of the microcontroller and its output wired to an input of the microcontroller. In addition to testing the circuit in the “digital domain”, you will use the analog input functionality of the microcontroller to measure the analog voltage level of the digital signals. The interaction between the hardware circuit and the microcontroller will be controlled via your computer.

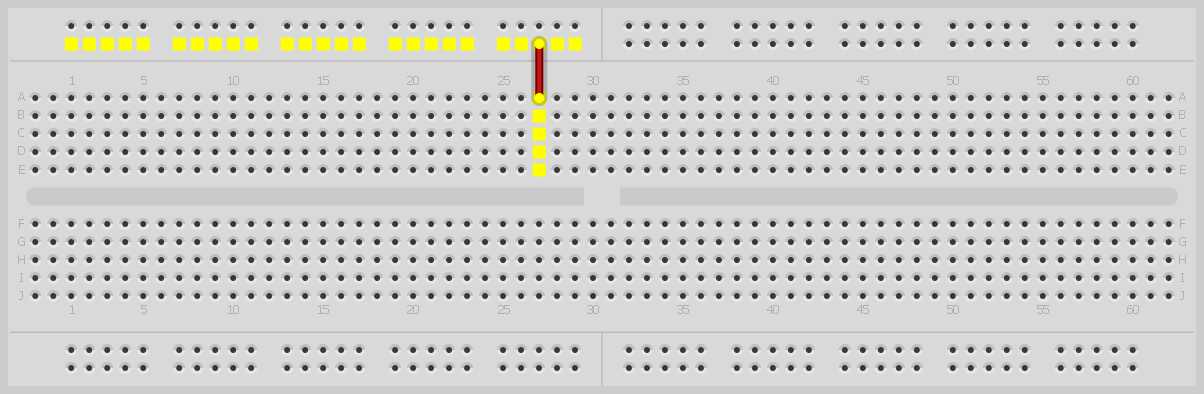
In the following chapters, you will be introduced to the breadboard, how to measure a voltage and how to use the graphical user interface to interact with the microcontroller.

You should have watched the “Hardware Kit – First Steps” videos and downloaded and installed the required software. The Firmata firmware should have been uploaded to the Arduino board and the Firmata control panel should be installed and able to communicate with the Arduino board before you start. If in doubt, refer to the video showing what to do in this lab.

**Breadboard Basics**

Breadboards are rapid prototyping devices that have terminals connected in a pre-defined way, allowing you to make connections between chips and wires by simply plugging them into the appropriate locations. A typical breadboard is shown in Figure 1.

The main breadboard area consists of terminals that are connected vertically in form of columns. The columns are separated in the middle by a spacer, called “gutter”. Integrated circuits (ICs) can be placed on top of the gutter, providing access to the IC pins via the terminals. To connect different columns, wires are being placed into the individual terminals. Breadboards usually have additional terminals on the top and bottom, which are connected horizontally. These areas are called “bus bars” and are used to distribute signals that are often reused throughout the circuit. The main use for the bus bars is the distribution of Power and Ground to all ICs in the circuit. Figure 1 highlights the connections of the bus bars and the main breadboard area in yellow, once they are connected via a wire.



**Terminal connections**

**Bus bars**

**Gutter**

**Wire**

Figure 1: Typical Breadboard with bus bars. Connected terminals are highlighted. (Image generated using Fritzing (www.fritzing.org).)

**Simple Logic Circuit**

Let's use the breadboard to build a simple logic circuit. Consider the simple logic function: Y=A•B•C. You can either use a 3-input AND gate or connect two, 2-input AND gates as shown in Figure 2.

**B**

**A**

**C**

**G = A•B**

**Y = A•B•C**

**B**

**A**

**C**

**Y = A•B•C**

Figure 2: Simple logic circuit realizing a triple input AND gate using two dual input AND gates.

Since digital logic circuits are used in a wide range of applications, integrated circuits exist which realize the basic digital logic gates, such as AND, OR, NOT, XOR, NAND and NOR. The “classic” family of digital logic circuits is the 74xx series, where the x’s stand for a number that defines the type of logic gate or circuit. Table 1 lists some of the most common elementary logic gates and their respective chip numbers.

|  |  |
| --- | --- |
| IC number | Function |
| 7400 | Quad 2-input NAND |
| 7402 | Quad 2-input NOR |
| 7404 | Hex NOT |
| 7408 | Quad 2-input AND |
| 7432 | Quad 2-input OR |
| 7486 | Quad 2-input XOR |

Table 1: List of Elementary Logic Gates within the 74xx series.

Traditionally, the (small) silicon chips containing the integrated circuits have been packaged in a “Dual in-line” package (DIP or DIL) with typically 14 or 16 pins. The package is called Dual in-line because it features two rows of pins with a spacing that fits across the gutter of a breadboard. Figure 3 shows a typical DIP-14 package. The package typically has a notch on one side or a marking on one corner. The marking indicates Pin 1. If you hold the chip so that the notch is towards your left and the pins are pointing down, then the pins are numbered on the near side from left to right starting from 1, then continuing with the pins on the far side from right to left (counterclockwise). Figure 3 shows an example of a DIP-14 chip with the corresponding pin numbers.



Since it is easy to integrate more than just one logic gate on the same chip, even the most basic logic ICs feature multiple gates per package. The traditional DIP packages contain four 2-input (or dual input) gates. Just as a side note, a few companies recently released single logic gate ICs in tiny surface-mount packages. These, however, are not breadboard-compatible and require etched printed circuit boards for interconnects, so we will stick with the traditional ICs.

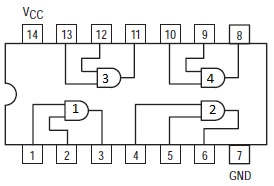
To gain experience in wiring digital logic circuits, let’s look at realizing the 3-input AND gate using the combination of 2-input AND gates as shown in Figure 2. First of all, we need to know which pin of the IC is internally connected to the AND gate(s) and which of the pins has to be connected to power and ground. To find this out, the manufacturer provides a so-called “datasheet”. Nowadays, it is easy to find these datasheets online. In our case, we will be looking for the datasheet for a 74LS08 IC, which contains four dual-input AND gates. The connection or “pinout” diagram is shown in Figure 4. For the “traditional” elementary logic gates within the 74xx series, the ground is connected to the lower right pin, while the positive voltage, which is called Vcc or Vdd for the HC series, is connected to the upper left corner pin.

Figure 3: Pinout diagram for a 74(LS)08 quad dual-input AND IC. Pins are pointing down. (commons.wikimedia.org/wiki/File:7408.jpg)

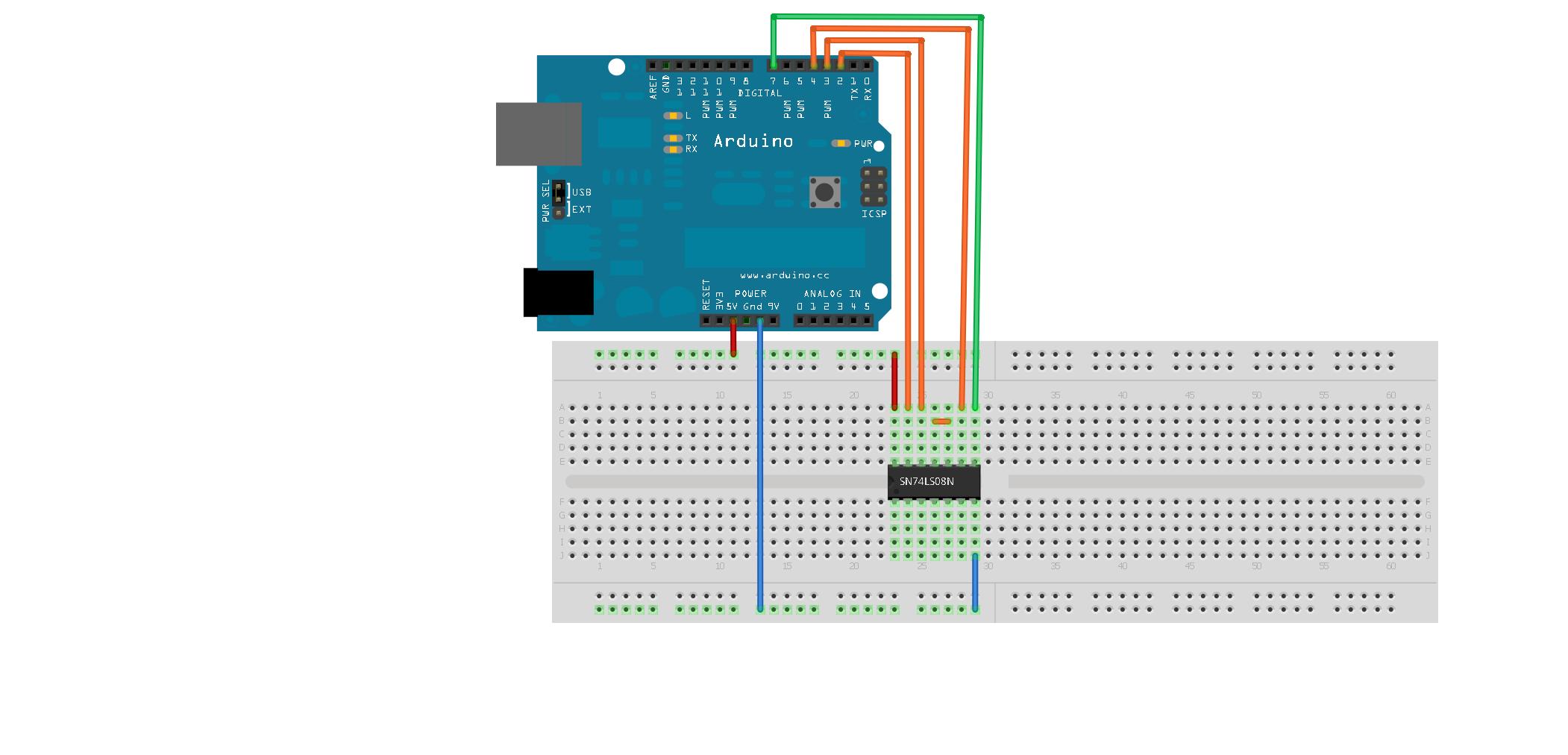
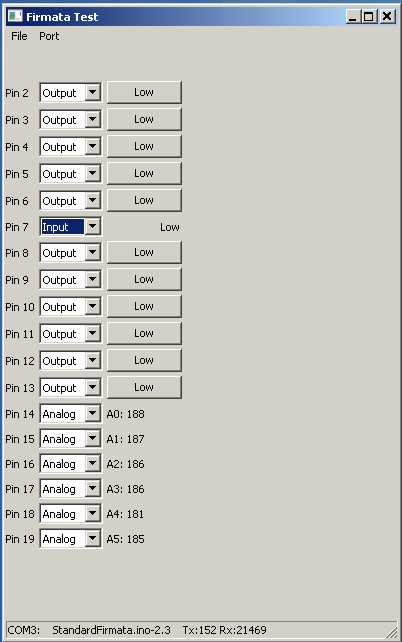
Figure 4: A 14-pin IC chip (Copyright of Motorola, Used with permission).

To complete the circuit, we will have to connect the output of an AND gate to the input of another AND gate. The IC is laid out in such a way that the output of an AND gate is next to the input of another AND gate, so we just have to “bridge” or “jumper” the respective pins using our breadboard.

**Task 0-1: Build the 3-Input AND Gate on a Breadboard and Test its Logic Operation.**

To build the circuit shown in Figure 2, follow the steps outlined below and refer to Figure 5.

1. Insert the 74LS 08 chip over the gutter as shown in Figure 5. Make sure that the notch is to your left. Although this is not mandatory, it tremendously helps figuring out the wiring and prevents accidental supply reversal, which the ICs do not like at all.
2. Connect the Arduino GND to a bus (Wire-1). Do not connect the Arduino +5V supply yet!
3. Connect the +5 V and ground busses to the appropriate chip pins. (Remember the buses are connected in a horizontal manner.) Connect the +5 V bus to pin 14 of the IC (Wire-2) and the ground bus to pin 7 of the IC (Wire-3) as shown in Figure 5[[1]](#footnote-1).
4. The circuit in Figure 2 has three inputs (A, B, and C). We will use three digital logic outputs from our Arduino board to control the values of these input variables. Let’s select Pin 2 for A, Pin 3 for B and Pin 4 for C.
5. The circuit in Figure 2 has one output (Y). We will use one digital logic input from our Arduino board to monitor the values of this variable. Let’s select Pin 7 for Y.
6. Connect the Arduino board to your computer and start the Firmata Test Panel.
7. Once the Firmata test panel starts up, select Pins 2, 3 and 4 to be outputs and Pin 7 to be an input. Set all outputs to “Low” as shown in Figure 5.
8. Connect Wire-4 from Arduino Pin 2 to Pin 13 on the IC.
9. Connect Wire-5 from Arduino Pin 3 to Pin 12 on the IC.
10. Connect Wire-6 from Arduino Pin 4 to Pin 9 on the IC.
11. Connect Wire-7 from Arduino Pin 7 to Pin 8 on the IC.
12. “Jumper” Pins 10 and 11 on the IC (Wire-8).
13. Connect the Arduino +5 V supply to a bus (Wire-9). After you do this, the entire bus is energized to + 5 V.[[2]](#footnote-2)
14. Test the circuit using the Firmata test panel on the computer. The indicator for Arduino Pin 7 should show “Low”. Now, set Arduino Pins 2-4 to “High”. If the circuit works correctly, Arduino Pin 7 should indicate “High”. Switching any of the Pins 2-4 back to “Low” should render Pin 7 “Low”. **Record your results on the Hardware Lab 0 Report Template**.



**VM**

**VM**

**B**

**C**

**A**

**Y**

**4**

**8**

**7**

**5**

**6**

**9**

**2**

**1**

**3**

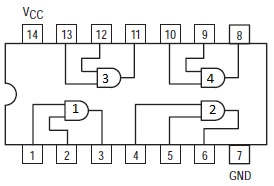


Figure 5: (Top Left) Firmata test panel used to interact with the digital logic circuit. (Top Right) Breadboard wiring example for 3-AND circuit. (Generated using Fritzing.) (Bottom) Pinout diagram for a 74(LS)08 quad dual-input AND IC. Pins are pointing down. Reincluded here for your convenience.

**Task 0-2: Use an Analog Input as a Voltmeter**

If there is no obvious short circuit, but the circuit is not working as expected, it is time for logic “debugging”. Odds are that one of the wires is not connected properly or the chip slipped out of the breadboard. But how can we test for an “open” connection?

The Atmel Microcontroller on the Arduino board has 6 analog inputs that can measure signals between 0V and +5V, which is ideal for our purpose. However, the analog inputs have a drawback: The Firmata test panel only displays a number between 0 and 1023 and not the actual voltage value. The number 0 corresponds to 0V and the number 1023 corresponds to +5V. We can solve this problem easily using a calculator and determining the actual voltage value by dividing the value displayed by 1023 and multiplying it by 5V.

To use the analog input as a voltmeter, make sure that Arduino Pin 14 is switched to “Analog” using the Firmata control panel as shown in Figure 5. The field “A0” on Firmata should show a number between 0 and 1023.

Next, plug a (flexible) wire into Arduino “Analog In A0”, similar to the yellow wire labeled “VM” in Figure 5. This will act as your “probe” to measure voltage levels in your circuit. Now, connect the other end of the “VM” wire to the following pins and record the numeric values indicated by Firmata for Analog Input A0 and **record the numeric values in the table** **on the Hardware Lab 0 Report Template:**

* GND bus
* +5V bus
* Unused (open) input of the IC, for example Pin 1 of the IC
* Unused (open) output of the IC, for example Pin 3 of the IC
* “Low” and “High” input of the IC. Connect the wire to Pin 9 of the IC and switch Arduino Pin 4 to “Low” using the Firmata Test Panel and record the numeric value indicated in Firmata for A0. Next, switch Arduino Pin 4 to “High” using the Firmata Test Panel and again record the numeric value indicated in Firmata for A0.
* Y output of the IC (Pin 8), recording the values for logic “Low” and a logic “High” indication by switching Arduino Pins 2-4 to “Low” first and then all to “High”.

You should notice that the unused or open input pins show a voltage value between a logic “Low” and a logic ”High”. This is a good indication if a connection to a gate input is not properly wired.

**Task 0-3: Take a Photo of your Completed Circuit**

Before you rip everything apart, take a photo of your completed circuit and attach it to the lab template. This will serve as a replacement for the “attendance stamp” that the in-person students have to get on their paper lab submission.

1. It is good lab practice to run wires around the sides of the IC rather than over the top of the IC. This will allow you to easily access the pins of the IC and enable you to replace the IC if it proves to be defective. [↑](#footnote-ref-1)
2. If there is anything wrong , for example the Power LED on the Arduino is not lighting up, communication to Firmata is not possible or the computer shows an error message, immediately unplug the +5V supply (Wire-9) and visually test your circuit for eventual short circuits between +5V and ground or a wrong wired IC. If you accidentally short the +5V to ground, protection circuits will prevent hardware damage, but the operating system might disable the USB port. Restart of the operating system might be required after that. [↑](#footnote-ref-2)