

Experiment M1 Review of Electronics Procedure

Deliverables: demonstration of working systems to Lab TA, checked lab notebook

Recommended Reading: Section 6.4; Chapter 18 of the textbook

You do NOT have to write a tech memo for this lab. Just make sure the TA fills out the score sheet as you complete each part of the lab. **Each item on the score sheet must be completed before the end of lab for you to receive credit for it.**

Overview

In this lab, we will review basic electronic circuits, instrumentation, and microcontrollers—topics that were covered in AME20216 – Lab I.

A microcontroller is a rudimentary computer packed into a small IC that can be programmed to automate various tasks. The Arduino UNO is an inexpensive microcontroller commonly used by hobbyists and engineering students. Shown in Figure 1, the UNO is connected to a breadboard where various sensors and circuits can be implemented (i.e. a thermistor in a voltage divider circuit).

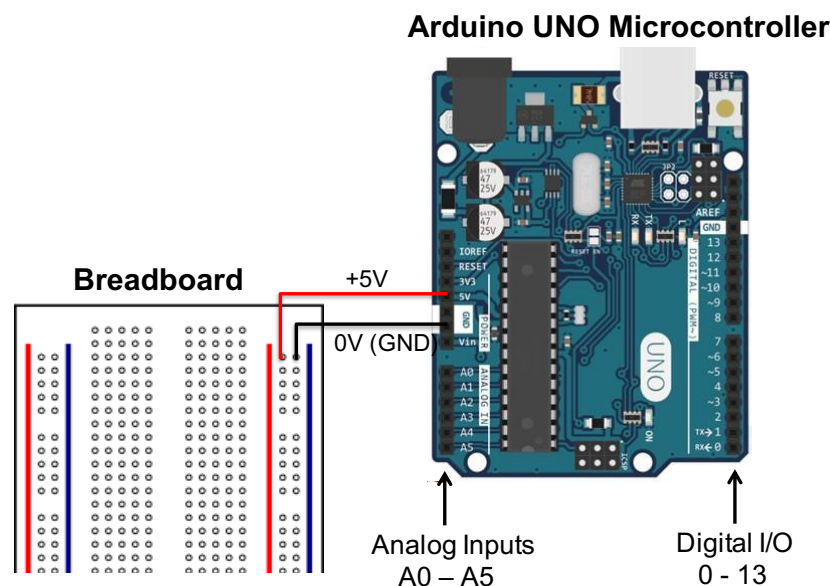


Figure 1 – The Arduino UNO Microcontroller works in tandem with breadboard.

Part I: Electromagnetic Forces

Background

Most mechatronic actuators use an electromagnetic force generated by a coil of wire to make things move. In this part of the lab, you will create a simple solenoid magnet by wrapping wire around a steel nail. The magnetic field of a solenoid is given by

$$B = \frac{\mu N}{L} I, \quad (1)$$

where $\mu = 1.26 \cdot 10^{-4} \text{ N/A}^2$ is the magnetic permeability for steel, N is the number of turns of wire around the cylinder, L is the length of the cylinder (not the wire), and I is the current.

The magnetic force F is proportional to the magnetic field B . Thus, the force created by a solenoid is proportional to the current I . You will experimentally observe how the current effects the magnetic force.

Procedure

1. Cut off a 32" long piece of enameled copper wire (38 AWG). Gently sand off the enamel insulation from the ends of the wire.
2. Measure the resistance of the wire. It should be a few Ohms. Record the value in your lab notebook. $R = ?$
3. Use a pair of calipers to measure the diameter of the steel nail. Record the value in your lab notebook. $D = ?$
4. Carefully wrap the wire around the nail to create a solenoid electromagnet. Try not to overlap the wire.
5. Estimate the number of turns N in the magnet using the length of the wire and diameter of the nail. $N = ?$
6. Set up the Keysight U8002A DC power supply:
 - a. Press the "OVER VOLTAGE" button and make sure it is set to 30V. Press the "OVER VOLTAGE" button again to save the setting.
 - b. Press the "OVER CURRENT" button and make sure it is set to 3 A. Press the "OVER CURRENT" button again to save the setting.
 - c. Plug the banana to grabber cables into the + and – outputs. Do NOT connect the magnet yet.
 - d. Press the Output ON/OFF button. You should see a value for voltage and current show up on the screen.
 - e. Toggle the Voltage/Current button and set the voltage to 0V.
 - f. Press the Output ON/OFF button. The screen should say "OFF".
 - g. Connect the cables from the "+" and "-" terminals of the Keysight U8002A DC power supply to the magnet wires using the minigrabbers.
7. Press the Output ON/OFF button. Gradually increase the voltage to 0.5V. Record the voltage and current in your lab notebook.

8. Try to pick up a paper clip with the magnet. Is the magnetic field strong enough?
9. Adjust the voltage on the power supply to various values between 0 and 3V. How much current is needed to pick up a single paper clip? $I_{pc} = ?$ Record the value in your lab notebook.
10. Use Eq. (1) to calculate the magnetic field B needed to pick up the paper clip. Record the value in your lab notebook
11. Disconnect the solenoid from the Keysight DC power supply, and connect it to the handheld DMM.
12. Tap the head of the nail on a magnet, and use the DMM on the 200 mV setting to measure the electromagnetically induced voltage. $V_L = ?$ Record the value in your lab notebook.

Part II: Digital Flip-Flop

Background

In this part of the lab, a push button will be used to toggle two separate LEDs between ON and OFF. The `digitalRead()` function will be used to determine when the button is pushed, and the `digitalWrite()` function will be used to light up either a yellow or a green LED.

digitalRead() – This function reads a voltage input, and returns a single bit as either HIGH or LOW. The bit is HIGH if the voltage is greater than 1.5V, and LOW if the voltage is less than 1.5V.

digitalWrite() – This function outputs a voltage that is either HIGH (5V) or LOW (0V).

Shown in Figure 2, the digital input pin 2 of the Arduino reads the voltage from the push button and stores it in memory as a single bit that is either HIGH or LOW.

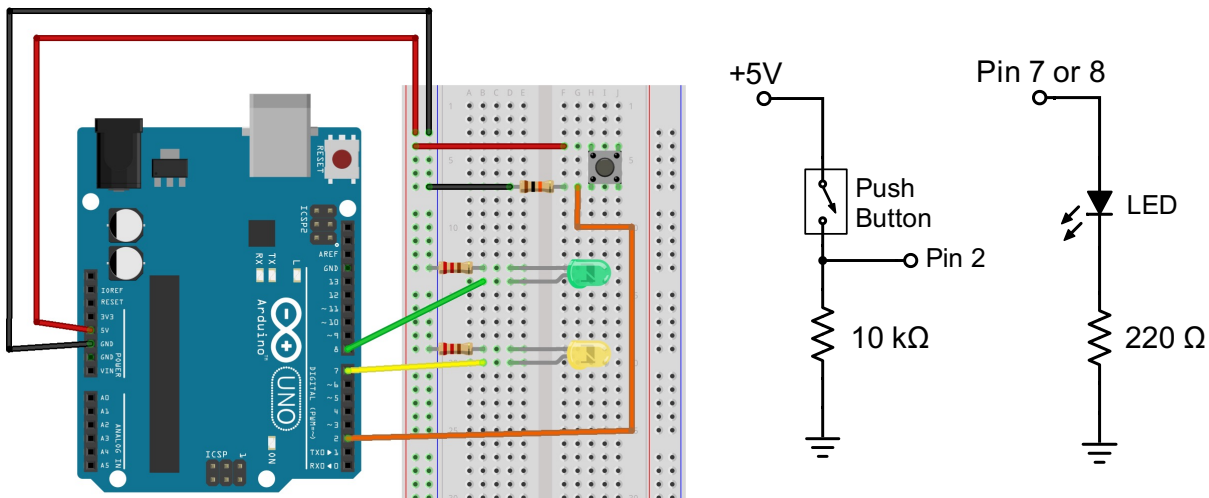


Figure 2 – A signal from a pushbutton is read into the Arduino via pin 2. Pushing the button toggles an internal state variable in the Arduino memory. The Arduino then illuminates either a yellow or green LED depending on the digital state.

Code Library

Throughout the semester, you will build a “code library” containing all of the Arduino Sketches and data sets that you generate. You must keep a multiple copies of the library on the lab computer, a thumb drive, or on the cloud.

1. Create a folder on the desktop titled “AME30358_YourNames”.
2. Create the following sub-directories (folders inside the folder you just created): “M1”, “M2”, “M3”, “M4”, “M5”, “M6”, “M7”, “M8”, “M9”, “M10”, “M11”, and “M12”.
3. Save multiple versions of your codes and data sets in the appropriate folder.
4. At the end of each lab, you must create back-up copies of this on your thumb drive or the cloud (i.e. Google drive or Box).

Procedure

1. Connect the Arduino UNO to the lab computer via the USB cable. You should see a green LED light up on the Arduino.
2. Use red and black jumper wires to connect the +5V and GND pins on the Arduino to the vertical bus lines on the breadboard, as shown in Fig. 2. Use the orange handheld DMM to verify that it is providing +5V of power.
3. Insert the push button into the breadboard such that its pins are in separate rows. Use a 10k Ω “pull-down” resistor to connect one pin to ground, and connect the other pin to 5V, as shown in Fig. 2.
4. Measure the voltage across the resistor. When you push the button, the voltage should go up to 5V (HIGH). When you release the button, the voltage should be pulled down to 0V (LOW).
5. Connect the output of the push button to digital I/O pin 2 on the Arduino.
6. Connect the Green LED in series with a 220 Ω resistor to ground. Connect the other end of the LED to digital I/O pin 8, as shown in Fig. 2. Make sure that the shorter “cathode” wire of the LED is on the ground (negative) side of the circuit.
7. Connect the Yellow LED in series with a 220 Ω resistor to ground. Connect the other end of the LED to digital I/O pin 7, as shown in Fig. 2.
8. Create a **code library**. Make a new folder on the desktop titled “yourName_code_library”. You will recycle and build upon your codes throughout the semester.
9. Download the “Part II Template” code template from the lab webpage, read the comments, and modify it.
 - a. Right-click the link and “Save as...” with an intelligent file name (i.e. “M1_FlipFlop_yourName.ino”). Save it to your code library.
 - b. Open the file in the Arduino IDE software.
 - c. Replace the *** in the beginning with the correct digital I/O pin numbers for the button and LEDs.
 - d. Replace the *** in the if-then structure at the end with HIGH or LOW, so that it does what the comments say it should do.

10. Use the Arduino IDE software to compile the code and send it to the microcontroller.
 - a. Go to “Tools” > “Board” and make sure either “Arduino/Genuino” or “Arduino UNO” is selected.
 - b. Go to “Tools” > “Port” and select the COM port that says “(Arduino/Genuino Uno)” next to it.
 - c. Click the check mark at the top of the Arduino program to check the code for errors.
 - d. Press the arrow button to compile the program and send it to the Arduino.
 - e. Go to “Tools” > “Serial Monitor” (or press “Ctrl + Shift + M”) to view the output from the “Serial.print()” commands at the bottom of the screen. (Make sure the Baud rate is set to 9600.)
11. Press the button on the breadboard. You should see the Green LED and the Yellow LED alternate between ON and OFF. Additionally, the serial monitor on the computer screen should update the status of the system each time the button is pressed.
12. **Demonstrate the working system to the TA, so you can be awarded points on your score sheet.**
13. Leave the +5V and GND connections plugged into the breadboard. Remove everything else.

Part III: Pulse Width Modulation (PWM)

Background

In this part of this lab, you will use a potentiometer as a knob to control the brightness of an LED. A potentiometer is a variable resistor whose value can be changed by turning a knob. Shown in Figure 3 below, the relative resistance between A and B and between B and C changes as the knob is turned. The potentiometer is essentially a variable voltage divider. Connecting +5V to terminal A and 0V (ground) to terminal C creates a variable voltage source on terminal B.

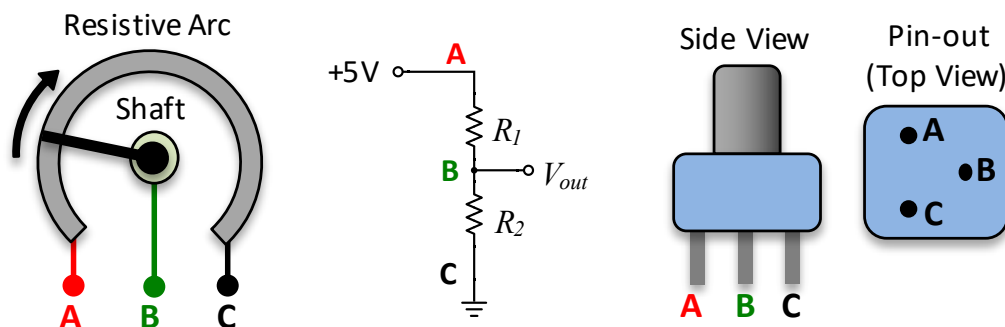


Figure 3 – (Left) A schematic representing the operation of a potentiometer. (Center) The potentiometer used as a voltage divider. (Right) The potentiometer has three pins A, B, and C. Pin B is sometimes called the “wiper”.

The potentiometer can be used as a variable voltage source. However, it is not capable of producing large currents, because the resistances of the potentiometer are fairly large, typically over 1000 Ohms. Thus the potentiometer cannot be used to directly control the brightness of the LED. To overcome this, we will use the Arduino microcontroller as a digital amplifier.

Shown in Figure 4, the analog input pin A0 of the Arduino reads the voltage from the potentiometer and stores it in memory as a 10-bit integer between 0 and 1023. This 10-bit integer is then linearly mapped to an 8-bit integer between 0 and 255, which controls the duty cycle of the PWM signal from pin 9. (In the code, the “analogWrite” function maps the integer 0 to 0% duty cycle, and 255 to 100% duty cycle.) The LED is connected to this digital output, and its brightness can be logically traced back to the angle of the potentiometer knob.

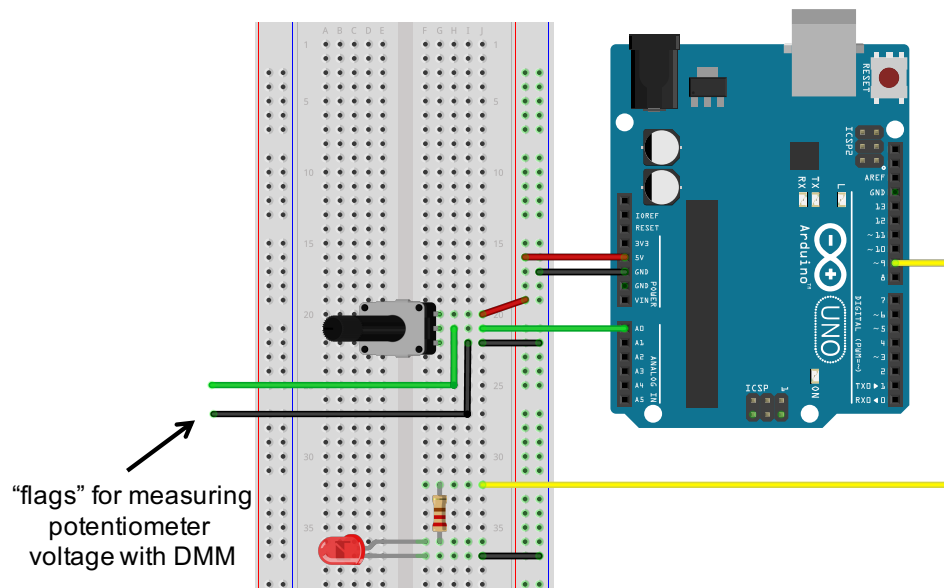


Figure 4 – The potentiometer and LED are mounted in the breadboard and connected to the Arduino.

Procedure

Important: You will use the Arduino to power the circuit on the breadboard. Do NOT turn on the breadboard.

1. Connect the Arduino UNO to the lab computer via the USB cable. You should see a green LED light up on the Arduino.
2. Use red and black jumper wires to connect the +5V and GND pins on the Arduino to the vertical bus lines on the breadboard, as shown in Fig. 4. Use the orange handheld DMM to verify that it is providing +5V of power.
3. Before you plug the potentiometer into the breadboard, connect the DMM to pins A and B, and measure the resistance. Turn the knob, and you should see the resistance change.
4. In your lab notebook, write down the resistance when the knob is turned completely clockwise, and the resistance when the knob is turned completely counter-clockwise. Repeat this with pins B and C.

-
5. Measure the resistance between pins A and C. Does the resistance change for pins A and C when you turn the knob? Base on the illustration on the left side of Fig. 3, does this make sense?
 6. Insert the potentiometer into the breadboard near the top, as shown in Fig. 4. Make the following connections:
 - a. Pin A on potentiometer \rightarrow +5V
 - b. Pin B on potentiometer \rightarrow Analog input A0 on the Arduino
 - c. Pin C on potentiometer \rightarrow 0V (GND)
 7. Use the DMM to measure the output voltage from the potentiometer between pin B and ground. Turn the knob, and the voltage should change accordingly.

Pro-tip: Insert a jumper wires into the rows of the breadboard where you wish to connect the DMM mini-grabbers. (These wires are sometimes called “flags”.) DO NOT try to grab onto the pins of the potentiometer or the base of any wire already in the breadboard.
 8. Connect the LED and a 220Ω in series between digital output pin 9 and ground, as shown in Fig. 4. Make sure that the shorter “cathode” wire of the LED is on the ground (negative) side of the circuit.
 9. Download the “Part III Template” code template from the lab webpage. Read the comments and replace the *** in the beginning with the correct pin numbers for the analog input from the potentiometer and digital output driving the LED (i.e., “A0” and “9”).
 10. Save the code with an intelligent file name (i.e. “M1_potentiometer_yourName.ino”). Save it to your code library.
 11. In the Arduino IDE software, go to “Tools” > “Board” and make sure either “Arduino/Genuino” or “Arduino UNO” is selected.
 12. In the Arduino IDE software, go to “Tools” > “Port” and select the COM port that says “(Arduino/Genuino Uno)” next to it.
 13. Click the check mark at the top of the Arduino program to check the code for errors.
 14. Press the arrow button to compile the program and send it to the Arduino.
 15. Go to “Tools” > “Serial Monitor” (or press “Ctrl + Shift + M”) to view the output from the “Serial.print()” commands at the bottom of the screen. You should see an integer between 0 and 1023 corresponding to the potentiometer angle (or voltage) continuously printed. (Make sure the Baud rate is set to 9600.)
 16. Turn the knob. The printed integer values should change, as well as the brightness of the LED. Set the knob, so that the integer is about 500 and the voltage on the DMM is around 2.5V.
 17. Use the oscilloscope to measure the PWM signal that is driving the LED. Connect the mini-grabbers to the appropriate places in the LED circuit, and press the “Autoset” button at the top of the oscilloscope. You should see a square wave.
 18. Turn the knob, and observe how the duty cycle of the PWM signal changes on the oscilloscope. Turn the knob so the average voltage on the DMM is around 1 V, and sketch the PWM signal in your lab notebook. Repeat this for a potentiometer voltage of 4 V.

19. Demonstrate the working system to the TA, so you can be awarded points on your score sheet.

20. Leave the potentiometer plugged into the breadboard for Part IV.

Part IV: Design Challenge

Combine Parts II and III to create a system with two LEDs, where you can use the push button to select which LED brightness will be controlled by the potentiometer.

When you have this working, demonstrate it to the TA. You must finish before the end of your lab section. (It is not the end of the world if you do not finish in time. This part of the lab is only 2 out of 16 points, and you will still get a B if you finish everything else.)

Clean-up

To receive full credit, you must return the lab bench to its initial state.

- Disassemble the circuit. Disconnect the wires from the relays, Arduino, and breadboard.
- Return everything to its appropriate bin.
- Make a back-up copy of your code library on a USB thumb drive or on Google Drive. You will need these codes later for future labs.

Data Analysis and Deliverables

You do NOT have to write a tech memo for this lab. Just make sure the TA fills out the score sheet as you complete each part of the lab.

Appendix A

Equipment

Part I

- Enameled copper magnet wire, 38 AWG (Digikey part #1528-2336-ND)
- Keysight U8002A DC Power Supply
- 2 – 3” long steel nail
- 220 grit or finer sand paper
- Small paper clips
- Permanent Magnet

Parts II and III

- Arduino UNO Microcontroller
- 6ft USB cable
- 12” jumper wires with male pins
- Small breadboard
- Bag containing:
 - Push button
 - Yellow and Green LEDs
 - Small Potentiometer w/ breadboard pins, 10k, blue
 - Various resistors
- Extech Handheld Digital Multimeter
 - One red banana-to-banana cable
 - One black banana-to-grabber cable

Appendix B

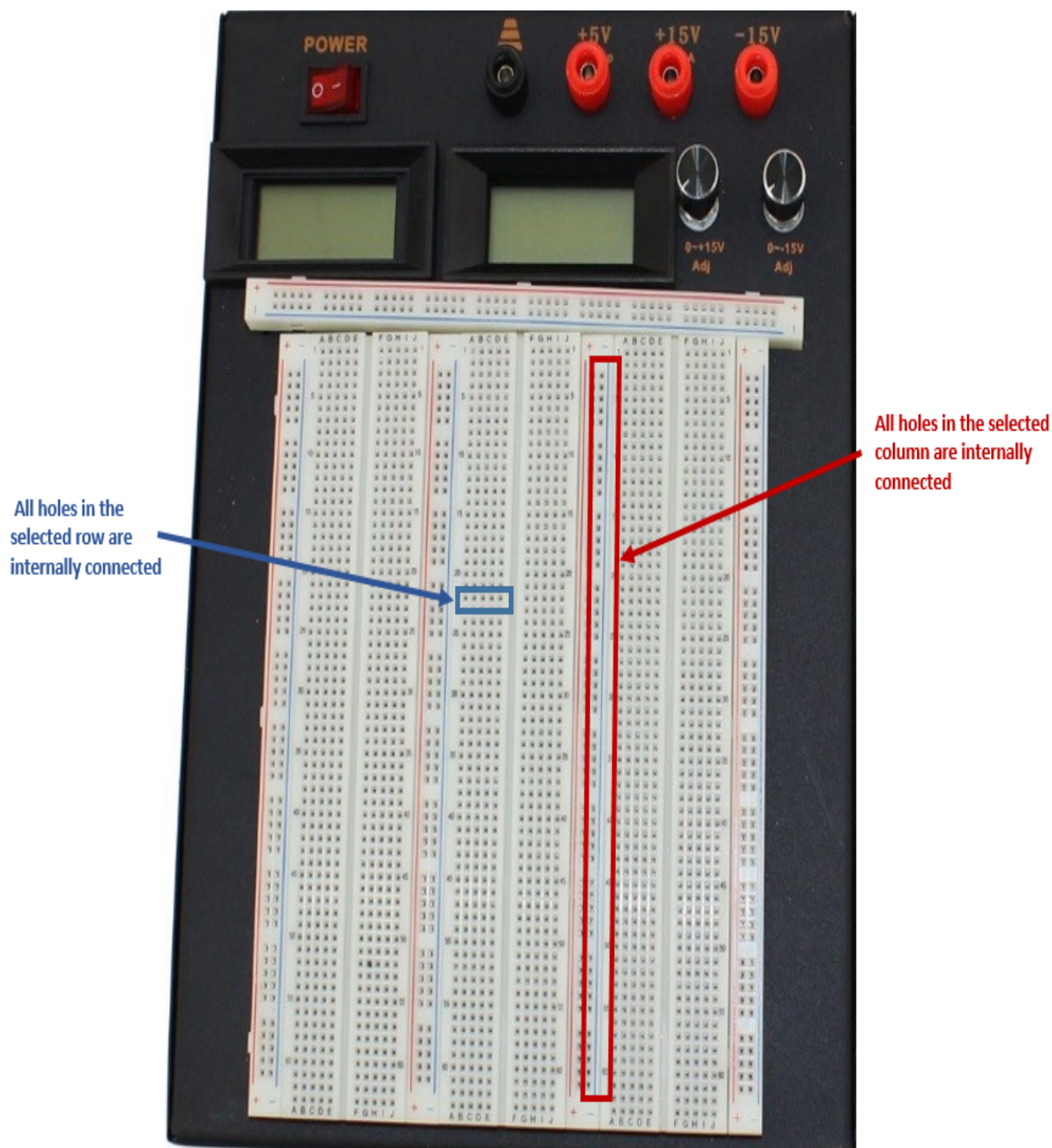


Figure 6 – Shown in blue, any 5 holes in a horizontal row are electrically connected, but they are NOT connected to the adjacent row of 5. Shown in red, all 50 holes in any vertical column or “bus bar” are electrically connected.