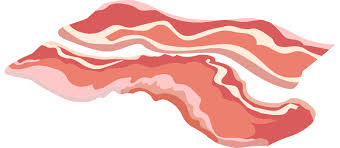
**The Unofficial Fetal Pig Study Buddy™**



Coded and Designed By:

Jeffrey Lupia

Jeffrey Lupia

30 November, 2017

CMPT 422: Comp Org. & Arch.

Professor Rivas

Abstract:

The Unofficial Fetal Pig Study Buddy™ was created to be used as a study tool for those who are studying the anatomy of the fetal pig. It is an interactive device which helps its user to memorize the names of the veins and arteries in a fetal pig. The Buddy™ is most useful in allowing its user to quiz themselves whilst physically looking at a diagram of the fetal pig’s circulatory system. The user needs only to touch the pointer to one of the metallic reference points, each mapped out to correspond to a vein of the pig, in order to get immediate feedback with the name of the vein or artery they selected. When selected, the vein or artery is displayed on a liquid crystal display attached to an Arduino Board for as long as the pointer remains touching the reference point. The device works by measuring various, unique voltages, each corresponding to a specific artery or vein. When a known voltage is read by the analog input of the Arduino, the matching Name is displayed on the Screen. The differing voltages are achieved by changing resistances across each reference point, subsequently affecting the voltage.

My Arduino project is titled *The Unofficial Fetal Pig Study Buddy™*, and is designed as a resource to aid students learning the names of the various blood vessels in a fetal pig. The fetal pig is a commonly used mammal across many introductory biology courses, as a way to teach students the basic concepts of anatomy and dissection. While the need for such a device may seem strangely specific to most, a biology student could likely understand the frustration experienced in trying to learn the scientific names of the internal anatomy of the fetal pig, especially when the only time they can be studied is when students are physically in the lab. When personally faced with this task of memorization, I was frustrated in the surprising lack of resources online. Apart from costly textbooks, a few less than ideal websites, and a very unhelpful iPhone app, my classmates and I were unable to find an efficient way of studying this topic. I recall several of my peers mentioning their frustration with the lack of resources available. This, and a subsequent class average of a 62 on the lab practical, inspired me to create the Unofficial Fetal Pig Study Buddy™, an interactive study tool for the Fetal Pig. I wanted to create a device that would allow students to quiz themselves easily on identifying objects in physical space, creating an experience more similar to that of taking a practical, than what one might experience simply reading and rereading labeled diagrams. Diagrams such as these can sometimes get confusing when different angles and differing perspectives are used, as they make it difficult for students to visualize in real space. The three-dimensional, hands-on perspective the Unofficial Fetal Pig Study Buddy™ gives, eliminates this issue, and allows students to work smarter, not harder. One of the inspirations for this project came from a small toy I recall having as a child. The toy was an interactive globe, with a pen attached to the base. When the pen was pressed to any country on the globe, an audio recording of the country’s name was played. The toy was meant to teach children the names and locations of each country in a way that they coud visualize more easily than a map. I have always found that people learn better when they engage with the information they’re studying, and the goal of the Unofficial Fetal Pig Study Buddy™ is to do just that.

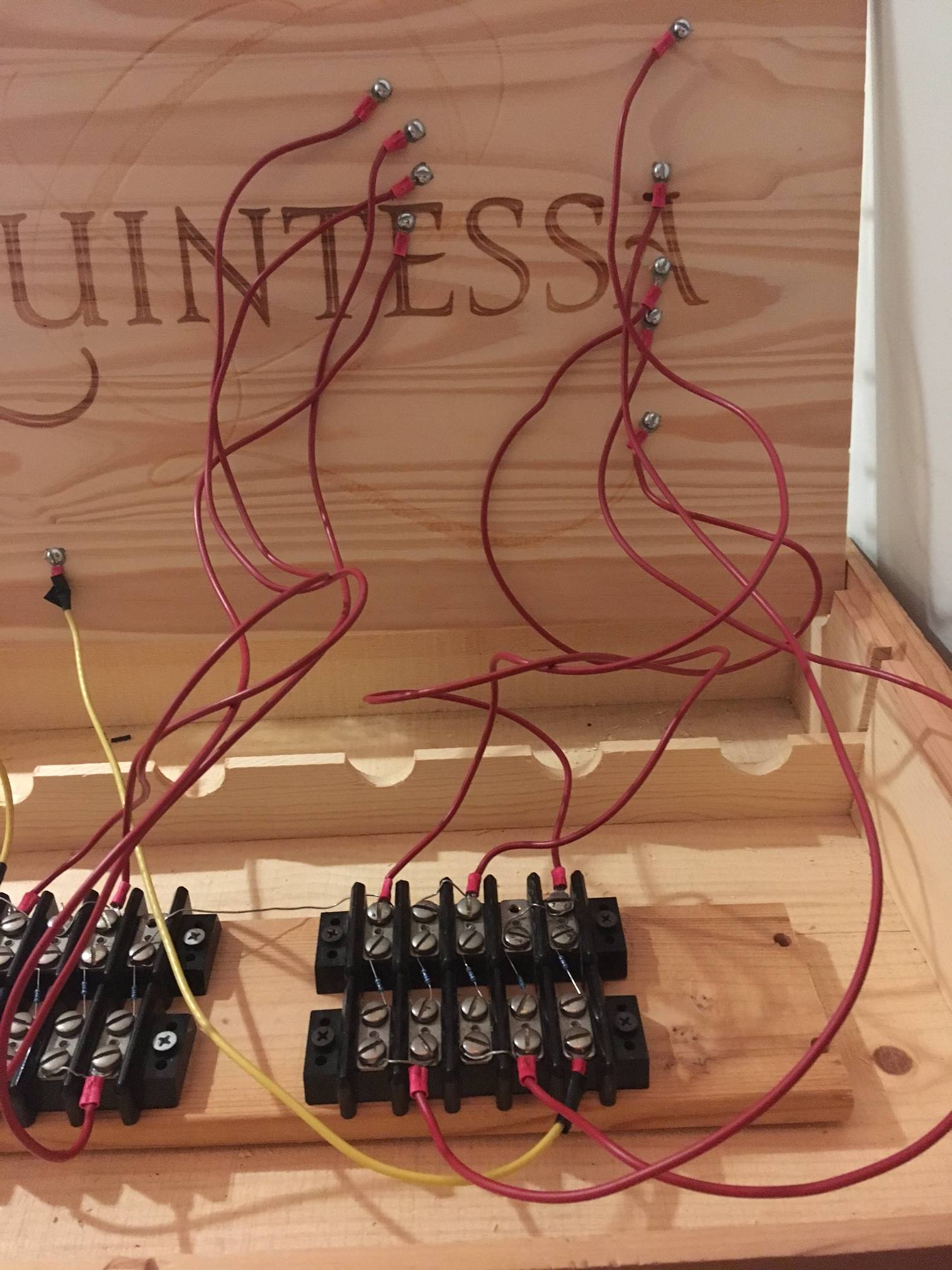
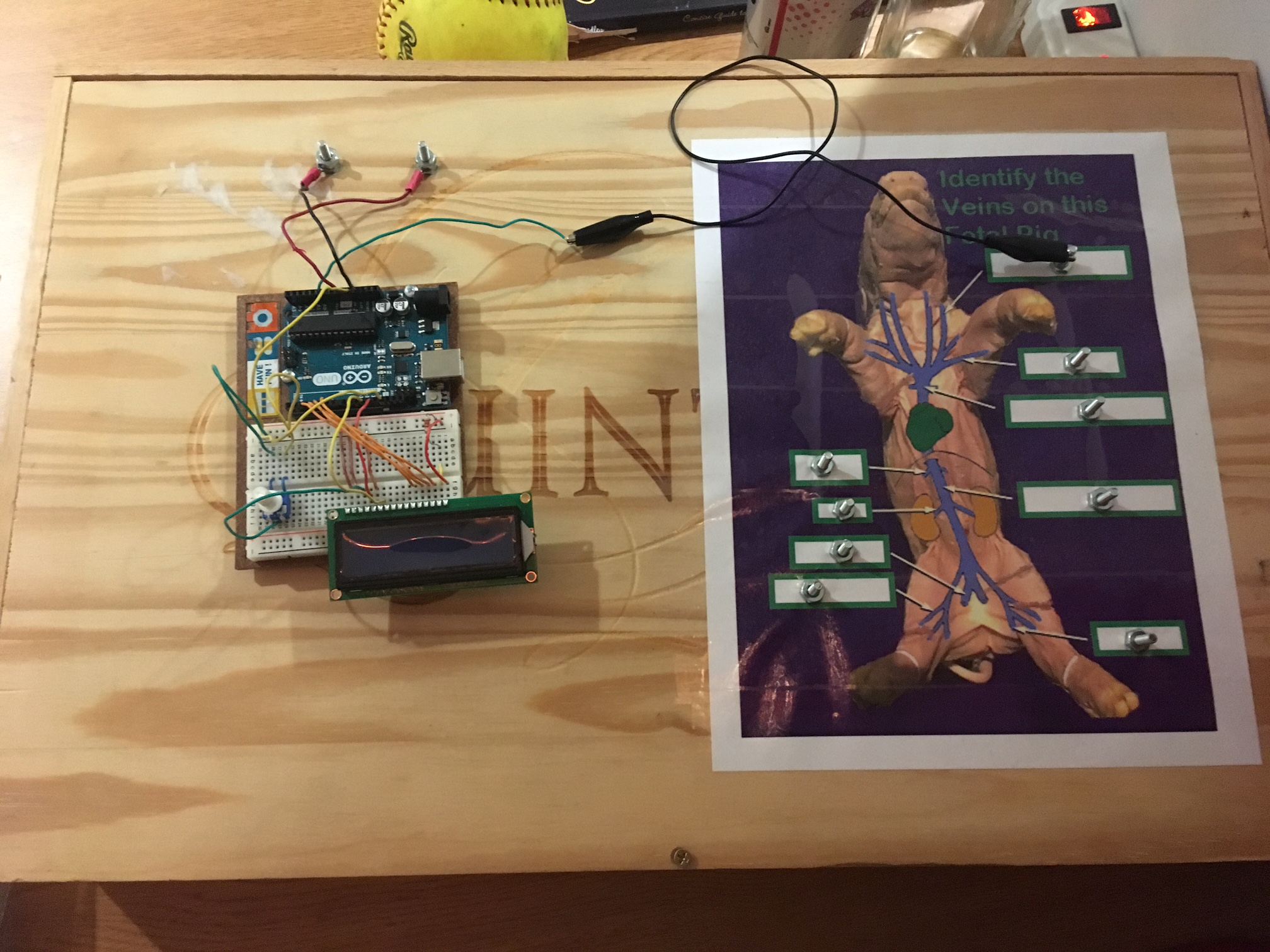
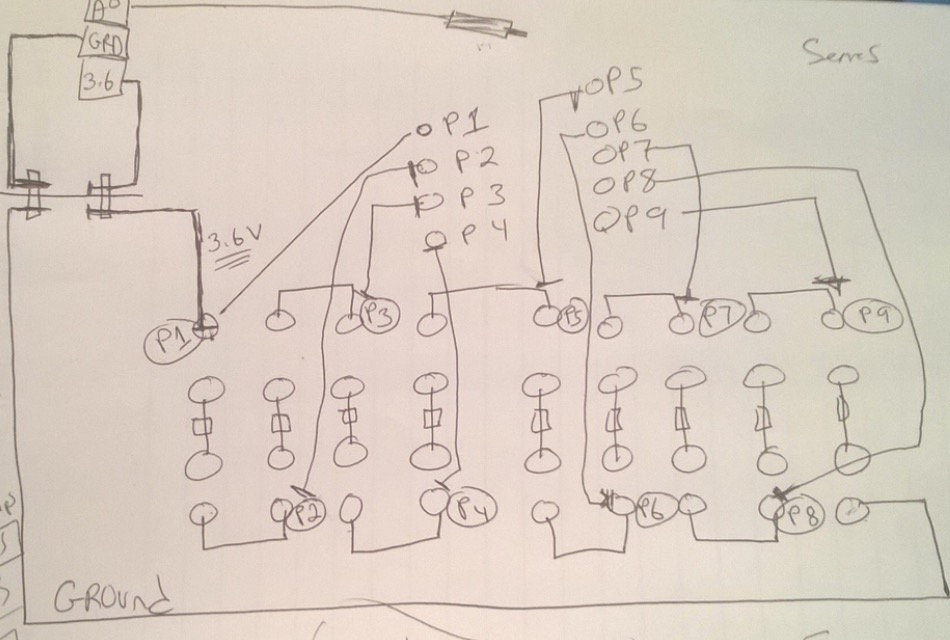
When I first began work on the Buddy™, I knew that I wanted to make it three dimensional, as it was meant to be something for people to physically interact with to study. I built the infrastructure of the device out of a wooden wine box, as I felt it would nicely contain any wiring inside, as well, as allowing me to pass electricity to the top of the Arduino through screws drilled through the wood. These screws were also useful in that they could hold down wires and connect circuits which the user is interacting with on the surface, to those on the inside. This would allow the project to seem visually nicer, by nicely containing the cables, and keeping as much of the hardware out of site as possible. I first added a hinge to the box to allow easy access to the wiring, and once it was functioning, it was time to determine how exactly the device would function.

The most important thing I needed to determine, was how I was going to make the Arduino Uno register that a specific vein or artery was chosen. My first thought was to have momentary push buttons representing each specific blood vessel, which could communicate back to the Arduino with a digital input. I quickly realized that this would be difficult, as the smaller Arduino parts would be difficult implemented nicely outside of the breadboard. Additionally, the Liquid crystal display, which I needed to display the names of the vessels were already taking up nearly half of the digital inputs, which would limit the amount of veins I could represent on the device. Instead, I decided to keep the idea behind the momentary switch, being the momentary completion of a circuit, however, I wanted to implement it differently. Eventually, I decided to make a handheld pointer which could attach to the Arduino, and be touched to reference points on the diagram of the fetal pig, in order to close the circuit. This circuit would be wired back to the 3.3 Volt power supply on the Arduino, as I knew that the 5 Volt supply would be needed to power the liquid crystal display.

The next major challenge I had to face was figuring out how to get the Arduino to differentiate between each reference point, and determine which one was being touched. It was inefficient to have each reference point individually wire back to the Arduino, and I looked into what would be the best way for the Arduino to have more than one potential reading in one input. The handbook *Make: Getting Started with Arduino* explained that the analog inputs of the Arduino have the ability to read what voltage is coming into it, and will subsequently map it to a 10-bit value, assigning a number from 0 to 1023 to represent each specific voltages ranging from 0 to 5 volts (Banzi, 66). I realized I could use one analog input to read all the reference points, and each could be differentiated by the output which it reads. In order to do this, I connected wires in a series, as shown in the diagram, with one end connecting to the 3.3 volt power of the Arduino, and the other end running the ground of the Arduino. The series contains nine 100 Ohm resistors placed at different points in the series. The reference points were all connected to the series at a different point, with each reference point completing a circuit, so that each complete circuit would have to travel through one more resistor than the last. The varying resistances allowed the analog input to differentiate between the reference points, based on the reading of how much voltage could get through. The pointer which the user touches to the reference points is directly wired into the analogue input, allowing the circuit to be completed, and the voltage read upon touch.

While working on this project, I ran into difficulties in both coding, and wiring. The most difficult challenge I ran into while trying to wire my project was the question of how to differentiate between reference points. I went through several stages of trial and error when working on this part of the project, as I had initially tried using resistors which were too strong, and would not allow enough voltage through. Additionally, I ran into an issue in which my original design of the resistors was in a parallel circuit, rather than a series. Once these had been figured out, the only other issue I ran into with wiring was initially forgetting to implement a ground for the circuit, which led to a lot of confusion before realizing what needed to be changed. On the coding side of things, the biggest difficulty I ran into was determining the values which each voltage read by the Arduino was being mapped to. On order to figure this out, I had to do some trial and error, by testing the values output through a serial monitor connection. Once I was finally able to determine the proper range for each, I used an if-else if statement in the coding to specify the different string outputs to the liquid crystal display. If the voltage read was within a range of values determined through testing, the program would output the name of the corresponding vein or artery, accomplishing what the project was intended to do.

The diagram below shows the final wiring design I drew out and used for the project. In addition to this, I used a liquid crystal display on my breadboard, set up based on the schematic found on Arduino’s website for their “*Hello World!*”example project, which is also shown below (Igoe). The breadboard is powered by the 5 volt pin on the Arduino, and connects to a potentiometer, which can control the contrast of the screen, as well as power its backlight. The digital outputs which run to it allow the display to exchange information with the Arduino, and receive the input given. I really enjoyed this project and find it very cool because it provides an additional resource for students like me who have trouble studying by simply staring at a picture with labels, and trying to retain the information.



Works Cited

1. Banzi, Massimo, and Michael Shiloh. *Getting Started with Arduino:* 3rd ed., Maker

Media, 2015.

1. Igoe, Tom, and Arturo Guadalupi. “‘Hello World!".” Arduino - HelloWorld, 9 July 2009,

www.arduino.cc/en/Tutorial/HelloWorld