Senior Design

Project Documentation

The Smart Digital Voltmeter

Sponsored by Commercial Lighting Enterprises Incorporated

Department of Electrical Engineering and Computer Science
University of Central Florida
Dr. Lei Wei

December 6, 2016

Group 19

William Brumby	Electrical Engineering	hunterbrumby1@gmail.com
Gaston Mulisanga	Computer Engineering	gtex1991@gmail.com
Travis Ram	Computer Engineering	trav.ram@knights.ucf.edu
Vladimir Tsarkov	Electrical Engineering	vtsarkov@knights.ucf.edu

Table of Contents

2 Product Description

3 Research

4 Standards and Realistic Design Constraints

5 Hardware and Software Design Details

6 Prototype Construction and Schematics

7 Prototype Testing Plan

8 Administrative Content

9 Product Operation

Appendix A - Resources

1 Executive Summary

A digital voltmeter is a tool which is used to measure either AC or DC voltage for a large range, usually from millivolts all the way up to kilovolts, depending on the product. Using two conductive probes attached to specified input jacks, the user can complete the circuit within the voltmeter itself by placing the probes in the desired location of the circuit under inspection. From the probes, an analog signal is passed through various safety components and circuitry to feed an analog-to-digital converter (ADC). This digital signal is now fed into a processor which computes and displays the data via LED or LCD display, depending on the model. The device generally has two input jacks, one for the positive lead and one for the negative lead which is the common ground to the supportive circuitry. The largest divide between modern digital voltmeters is whether the device is manual or auto-ranging, i.e. whether the processor can realize the range of value you are attempting to measure and select the correct path the signal must follow to be read correctly. Parameters such as these are detrimental in determining the direction in which our voltmeter design will progress.

The goal of this project is to develop a lightweight and compact digital voltmeter which wirelessly transmits data and measurements to an accompanied smartphone application. The data to be collected and measured by the meter will be both AC and DC voltage. The final product should provide an ease of use far greater than the current voltmeter technology where the user can record values to observe later. This voltmeter should be a low power device that can run for many hours (around 200 hours) so when the product is used in the field there is no concern of battery life. The user will be able to control settings such as sampling rates and measurement ranges via the application. The device will also include a small traditional display for times when smartphone access is inconvenient or unavailable. The smartphone application will allow users to view measurements over time and automatically generate plots of the data as desired.

This product is designed for engineers, electricians, hobbyists, and general household usage. Although most users prefer digital multimeters, devices that can measure several different electrical and physical parameters in a single package, some users only need to measure voltage in specific applications and want a device with high level accuracy and precision. We will be fully designing and creating a device that will precisely carry out these operations and present the data in a method not previously done before. The advantage to our device is the ability to record, access, and view these measurements via a smartphone application. The Smart Digital Voltmeter will give users the ability to record the data hands-free and save it automatically for later evaluation. Having the ability to control precisely what you desire to measure for any range is the reason our device, despite its limited functions, is far more beneficial to the average user. The device will conform to all safety and communication standards and constraints as defined in section 4. It should be noted that even though our product will conform to the safety standards (Category rating), we do not have the time nor money to obtain actual safety certifications.

2 Product Description

2.1 Motivation

As electrical and computer engineering students, our team has used voltmeters on countless occasions for lab work, as general electronic hobbyists, or at our real jobs. What we have all encountered and can all agree on when taking measurements is the inconvenience in logging data. Traditionally, measurements that are taken are able to be read by the user via the LCD or LED display on the device itself. The downfall to this outdated design is recording this information, due to the fact that when the probes are removed, the data disappears. Most modern voltmeters have the ability to “hold” the value displayed so the user may release the probes and record the information. While this is useful and partially identifies the issue at hand, the main problem still lies with the inconvenience in removing the probes, logging the information, and re-probing the circuit for another measurement. For example, let us consider a circuit designer who designs and constructs a low-pass filter and wants to test the output of his/her circuit. Ideally, an oscilloscope would be the desired tool to use in this situation. Unfortunately, these tools are very expensive, but the measurements can still be taken with a digital multimeter. The user would set the function generator on the input of the filter to a low frequency, probe the output, and read a voltage. The user must then remove the probes, record the value, set the next frequency, and re-probe the circuit. This can become very tedious when narrowing down specific parameters such as cutoff frequency, maximum overshoot, etc.

Our goal is to create a product that eliminates this inconvenience, along with aiding in the large-scale ability to decipher the data that would be easily done by an oscilloscope. This is where the idea of data being wirelessly transmitted to a smartphone application was born.

2.2 Goals and Objectives

The goals for our product are very simple. We want to design a basic voltmeter that can be used virtually anywhere with the ability to send and record data measurements via wireless transmission to a smartphone application where the data can be further analyzed. Functionality of the voltmeter will be limited to AC and DC voltage for a range specified in section 3. The measured voltage can be viewed on a digital display located on the meter itself, along with being wirelessly transmitted to the accompanied smartphone application. Users will have many options for recording and analyzing the transmitted data via this application. Some of these options include sampling rates, data plotting layouts, and many more settings. Once measurements have been taken, the user will be able to automatically plot the recorded data inside the application. Being able to take multiple measurements at a sampling rate determined by the user will prove to be the most valuable aspect of our product. The reason being our cheap and compact voltmeter can mirror the most basic function of a large and expensive oscilloscope, which is seeing the characteristic response of circuitry in nearly real time. This is done accomplished through a powerful processor and strategic design architecture which can sample input voltage at very high speeds. If users desire a single quick and accurate measurement without the need of the application, the digital display will be programmed in a way that mirrors modern voltmeters. Additionally, if the user wants to see other voltage parameters such as maximum, minimum, and average values over time on the smartphone application.

The PCB, electrical circuitry, and digital display will be contained within a small container which can be clipped to a belt or stood upright on a flat surface. The probes will be light and compact with comfortable grips and safety guards to ensure the user does not accidentally make contact with the circuitry being measured. Because we are designing an instrument to be potentially used with high current or high voltage measurements, safety is of the utmost concern. Physical high-voltage isolation slots will be embedded into the PCB, along with numerous safety measures such as fuses, varistors, thermistors, and diodes to aid in the protection of both the product itself and more importantly the user. Our final design will utilize high quality products such as ceramic fuses (in comparison to cheap glass fuses) and high power resistors to ensure the safety of both product and user in cases of transient voltage spikes.

2.3 Requirements Specifications

The following are the key design requirements and specifications of our smart voltmeter:

- Digital display that will display the current voltage being measured, updated at a rate specified by the user via the smartphone application
- Wireless capability of connecting the voltmeter to a phone or computer in order to record measured data and will comply with that wireless technologies standard.
- Two probes to measure positive and negative terminal.
    - Button on the positive probe to record measured values.
- Voltmeter will be lightweight and compete with current voltmeters.
    - Will weigh under 300 grams.
- The device will be operated as similar as possible to current voltmeters to retain user familiarity.
- Our voltmeter accuracy will be within 1% that of current voltmeters.
- The cost will be as low as possible to maintain competition with other digital voltmeters that do not provide wireless data recording.
    - Finished project is expected to cost under $100.
- The device will be low power in order to maximize battery life, since it will be a compact wireless data transmitter.
- Dimensions close to 8 x 4 x 2.5”.
- The Android application developed will operate in accordance with the standards set by Google to be available on the Play Store.

Although creating a digital Voltmeter may sound simple in theory, there are many requirements and specifications that must be met very carefully if this design were to compete with existing meters. Our voltmeter, as its name suggests, will provide accurate AC and DC voltage readings. In order to introduce a product, it must first stand a fighting chance against the current standard voltmeters that have been around for many years. Therefore, it must at least match the existing models to have a chance at success. Our smart digital voltmeter will be required to maintain a familiar feel to existing digital voltmeters. That is, a regular user of today's voltmeters should be able to switch over to our smart voltmeter with relative ease. Our design and utility should appear familiar and very similar in its functions as the average voltmeter. If our product will be too difficult, consumers will hesitate to make the transition to the smart digital voltmeter. Furthermore, the buttons, probes, and functions (AC and DC voltage) should all look and feel familiar.

In addition to function and familiarity, the smart digital voltmeter will be specified to weigh, cost, last, and be as accurate as the average voltmeter. The smart digital voltmeter will be required to have dimensions no larger than 8" x 4" x 2.5" and weigh no more than 300 grams. Any larger, and we run the risk of the voltmeter losing its portability advantage. Next, the smart digital voltmeter must be affordable and cost less or equal to the competition. Because our idea of recording voltage readings is close to the luxury category, consumers are not likely to invest their money into a luxury product if it is too expensive. Another key specification of the smart digital voltmeter is its battery life. The voltmeter will operate on battery power, and thus should have low power consumption in order to last as long as possible. With regular use, the smart digital voltmeter batteries should last approximately as long as the voltmeters from competitors (~36 hours). Last but not least, and one of the most important specifications of a voltmeter is its accuracy. We expect our smart digital voltmeter to be within 1% accuracy of the average handheld voltmeter.

Now this is where the smart digital voltmeter shines and has the edge over today's voltmeters. Wirelessly transmitted data recording of measurements is what will set the smart digital voltmeter apart from the rest. Likely to be an Android application, our software will allow the wireless transfer of data from the voltmeter onto an app on a smartphone. The user can then observe changes in voltage readings and share this data with others wirelessly. In other words, the application will serve as a small portable oscilloscope that can instantly share data and results with colleagues.

2.4 Quality of House Analysis

↑ = Positive Correlation

↑↑ = Strong Positive Correlation

↓ = Negative Correlation

↓↓ = Strong Negative Correlation

Table 2.1: House of Quality Trade-Off Table

Marketing Requirements	**Engineering Requirements**				
	Low Power	Accuracy	Low Cost	Dimensions	Lightweight
Accuracy	↓	↑↑	↓	↓	↓
Low Power	↑↑		↑	↑	↑
Ease of Use		↑	↑	↑	↑
Low Cost	↑	↓	↑↑	↑↑	↑↑
	Battery life at least 200 hours	Within 1% of standard voltmeters	≤$100	8 x 4 x 2.5”	≤300 grams

This device has potential to be used in many fields, such as engineering, people’s homes or any areas that need measurement relating to voltage, current, and resistance. On top of that, the device should be portable, easy to carry, and lightweight. Another vital factor, is that it will be used in remote areas where it cannot be connected to a power outlet for long periods. Because of the reasons mentioned above, the device will be designed such that it uses as less power as possible. A low power device goes hand in hand with lightweight device and easy of carrying around. Which will be very appealing to our target users.
Since this device will have multiple target users. We have taken into account that not all users will have technical expertise in design or circuitry, therefore for us designers, it is very important that the voltmeter should be easy to use for all users. The more user friendly, the more acceptable it will be. This correlates with the price of the device. It makes sense to buy a device that is easy to use for everybody.

None of the things mentioned above are important if the device is not working. Our first objective will be to make the device work and do what it is intended to do, that is measure AC and DC voltage. Not only must it work but it should display the output as accurately as possible, within 0.1% of standard voltmeters in the market. This will give us a competitive edge over other voltmeter makers. Overall our voltmeter will be a low cost device machine that gives accurate and precise display of the measurements taken. To make it economical and practical, it will be small, portable, and use low power consumption.

3 Research

Traditional analog voltmeters have been around since the 1920s, and the conversion to digital technology emerged in the mid-1950s. That being said, there is superfluous material to research to gain insight into the process of designing and implementing a modern digital voltmeter. In this section we will briefly discuss the background of the electrical technologies used in these products, along with the pros and cons of existing products. We will then discuss the strategy behind our parts selection and give a general idea on how our product will be implemented using these components.

3.1 Technological Foundation

As mentioned above, the first multimeter, which could measure voltage, resistance and current, was invented nearly a century ago. This device, called the Avometer, operated using moving coils, precision voltage resistors, rheostats, and switches and sockets to control range selection. It also incorporated a “universal shunt” resistor to measure seven ranges of current. Later models developed in the 1950s included the ability to measure AC current and voltage. Analog multimeters such as this one fared well in voltage and current measurements but had limited ability when measuring large resistances. Along with electrical limitations, analog multimeters are generally bulky and non-mobile. Like the rest of the world around this time, the conversion from analog to digital also struck accord with products such as the multimeter. While not as accurate in its early stages, the digital multimeter offered users a more compact and overall usable product. Digital technology provided a platform for advanced functionality in multimeters along with other improvements such as power consumption. These advancements are the reason we decided to incorporate digital technology in our device.

3.2 Existing Digital Voltmeter Projects and Products

In 2013, the engineering department of University of Petrosani in Romania proposed an application that also added extra functions of a digital multimeter. Their project was similar to ours in that it allowed a digital multimeter to connect to a computer and gather information from the multimeter via an application. This allowed for the user to observe and record data in real time. Data was recorded in a chart and in a table with appropriate time markers as measurements were performed. It is important to note that the key element of a digital multimeter is an analog digital converter (ADC). This allows every value of voltage to be represented by a string of binary code that is much more resistant to noise than in analog multimeters. Therefore, digital multimeters are more accurate than the analog counterparts. Although this engineering department of University of Petrosani had the same goals as our smart digital voltmeter, rather than designing their own multimeter, they simply used existing inexpensive models of multimeters and wrote software to integrate a digital multimeter with a computer.

On June 4, 2015, the editorial team from All About Circuits provided a project on how to make a simple digital voltmeter using an Arduino and 16x2 liquid crystal display (LCD). This project uses the advantage of Arduino's onboard analog to digital converter as well as several analog input pins that connect the positive and negative terminals of the probes. Arduino's ADC is a 10-bit converter resulting in 2^{10} or 1024 output values ranging from 0 to 1023. By using Arduino's 5V reference voltage, the team was able to calculate voltage present at the analog input. Additionally, this project applied a voltage divider in order to measure voltages above that of the reference voltage. The editorial team connected the input voltage that is desired to be measured to the analog input pin. The Arduino board then takes this analog value and multiplies it by the reference voltage (in this case it is 5V), then divides the product by 1024 to calculate the actual voltage value. After calculating the voltage, the measure value is recorded on the LCD screen.

Because of the limiting factor of the Arduino voltmeter being able to measure voltages of less than the 5V reference voltage, the editorial team from All About Circuits applied a voltage divider. The input voltage needed to be divided so that the voltage that actually went into the Arduino is 5V or less. For this project, the team used a 90kΩ and a 10kΩ resistor to create a 10:1 ratio voltage divider. In doing so, voltages up to 50V could be measured.

Another Arduino project by T.K. Hareendran created a digital voltmeter via Arduino that allowed voltage readings between 0V to 30V. Because Arduino's analog inputs can be used to measure DC voltage between 0V to only 5V, Hareendran used voltage divider to decrease the voltage being measured by the Arduino input. Code in the Arduino software is then used to calculate the actual voltage being measured by the hardware. The following is a schematic showing the voltage divider in action:

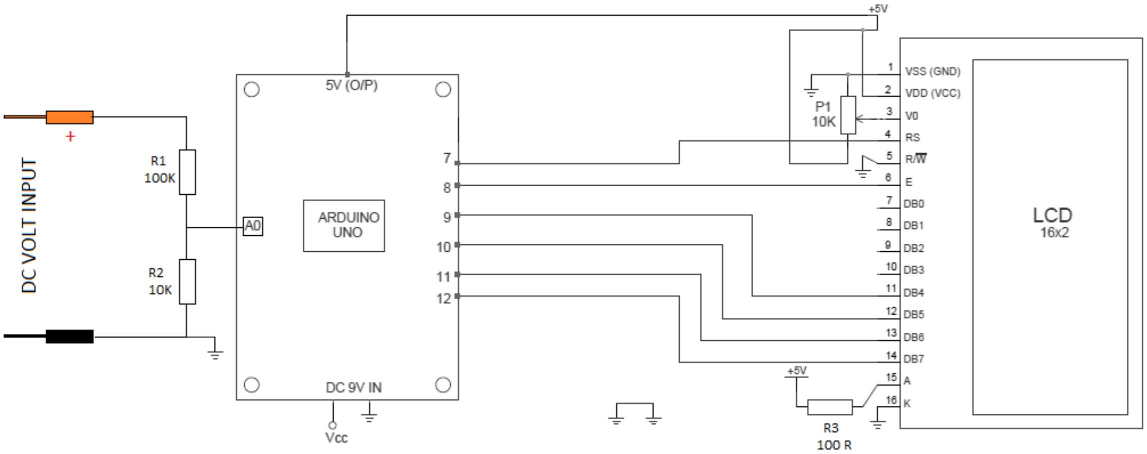


Figure 3.1: T.K. Hareendran's Project.

As can be seen from the above schematic, the two resistors used for the voltage divider are 100kΩ and 10kΩ. With these values, it is possible for the Arduino board to read voltages from 0V to 55V. The voltage being fed into the analog pin on the Arduino board is V_{in} [(10k) / (100k +10k)] which is approximately $0.09V_{in}$. Because this is the input voltage divided by 11, this means that the reference voltage of 5V multiplied by this coefficient of 11 yields 55. Therefore, 55V is the upper bound of what this Arduino voltmeter can measure. However, according to Hareendran, it is good practice to label a voltmeter such as this one a "0-30V Digital Voltmeter" in order to add a "safety margin."

Another significant portion of our project consists of Bluetooth® data transmission and reception. On the "Do it Yourself Hacking" website (DIYHacking.com), Mayoogh Girish created a simple project focusing specifically on Arduino Bluetooth® Interfacing with basic Bluetooth® functionality. In Girish's project, he allowed the user to control a light emitting diode (LED) on the Arduino board using a smartphone. The following flowchart shows how Girish's project functions:

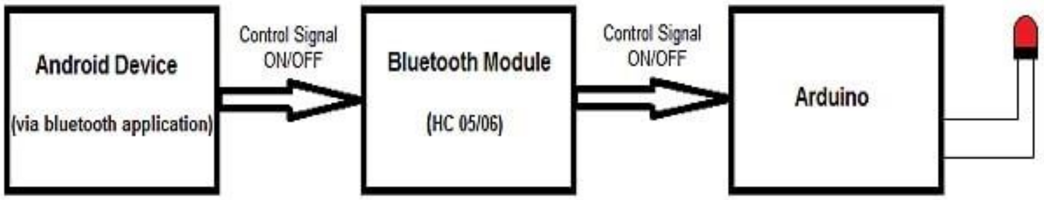


Figure 3.2: Mayoogh Girish's Project.

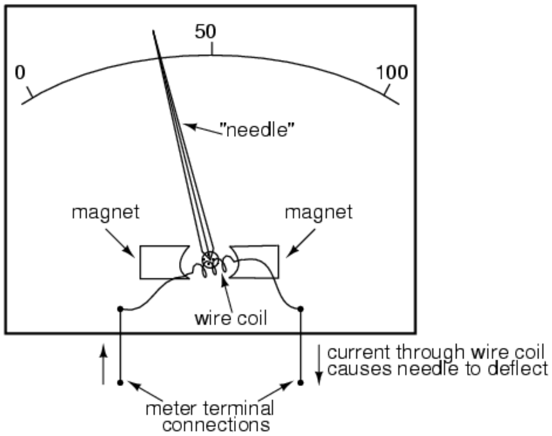
As can be seen from the above flowchart, this sample project can be broken down into three main parts: An Android smartphone, a Bluetooth® transceiver, and an Arduino board with an LED. The HC 05/06 Arduino Bluetooth® Module that is used in this design work on serial communication. In other words, only one bit can be transferred at a time. Based on this project, the Android app is designed to send serial data to the Arduino Bluetooth® module when a button is pressed on the app on the smartphone. The Arduino Bluetooth® module at the other end receives the data and sends it to the Arduino through the TX (or transmission) pin of the Bluetooth® module, which is connected to the RX (or receiving) pin of the Arduino board. Then, the code written into Arduino checks the received data. If the received data is 1 or HIGH, then the LED turns ON. If the received data is 0 or LOW, then the LED turns OFF.

3.2.1 Different Types of Voltmeters

There are several different types of voltmeters available that we will discuss and they each have a subcategory of DC Voltmeter, AC Voltmeter, or two in one.

1) Permanent Magnet Moving Coil (PMMC) Voltmeter - This type of Voltmeter is an analog voltmeter that uses two stationary magnets to generate a magnetic field with a wire coil in between the two magnets. These magnets are typically made up of alcomax and alnico materials in order to provide high field strength and increase reading accuracy of the voltmeter. The wire coil moves freely between the two permanent magnets shown at the left. This coil is wound in

Permanent magnet, moving coil (PMMC) meter movement



many turns of copper wire and is then placed onto a rectangular piece of aluminum which is connected with bearings to allow it to rotate. The higher the voltage that is being measured, the larger the current will be that is passing between the two meter terminal connections, and as a result, the higher the current, then the higher the voltage the needle will point to on the scale.

There are three main disadvantages of this type of Voltmeter. First, it is only capable of measuring DC voltage. If we apply AC current to a PMMC, the direction of current will be reversed half the time and as a result the torque will also be reversed, yielding an average torque of zero for the entire AC clock cycle. A second disadvantage of PMMC voltmeters, can be errors of the permanent magnets. Over time and due to temperature effects, magnets may lose their magnetism, resulting in inaccuracies of voltage readings. Third, temperature affects the resistance of the moving coil. Normally, the temperature coefficient of copper wire in the moving coil is 0.04 per degree celsius rise in temperature. Due to lower value of temperature coefficient, the temperature rises at a faster rate and causes the resistance to increase. A significant amount of error is caused due to this.

As for good news, PMMC has its advantages in power consumption. This type of device is very low power and can last a very long time. Additionally, PMMC voltmeters make it very easy to measure quantities and the scale is very easy to read and is known for its high accuracy.

2) Moving Iron (MI) Voltmeter - Moving Iron Voltmeters consist of two different types, repulsion type and attraction type instruments. These types of instruments are similar to the PMMC voltmeter due to the fact that they are all analog voltmeters. However, both MI Voltmeters have an advantage over the PMMC voltmeter because they can measure both DC and AC voltage.

- Repulsion Type MI - The basic idea behind the repulsion type MI Voltmeter is to use two separate iron strips in parallel with similar magnetic properties (same north and south poles). Due to the similar magnetic properties, the two plates will repel each other. This repulsion force depends on the strength of the electromagnetic field in the coil which surrounds the two plates. The iron strips will keep their repulsion independent of the current direction of the coil because the magnets will remain similar in polarity. Because of this, AC readings using this device are available. Next, one plate is placed in a fixed position so that it cannot move, while the other plate is connected at one end. In other words, both poles of strip 1 are fixed, while only one pole of strip 2 is fixed. This allows repulsion to take effect and for the unconnected pole end of strip 2 to move farther away from strip 1, proportional to the electromagnetic field in the coil and thus the current through the coil. Lastly, a pointer is connected to the free end of the iron strip 2, and a scale is located near this pointer to indicate the voltage reading based on the current running through the coil. This is possible because the force of repulsion is proportional to the current, which is proportional to the angle displacement of the pointer. Repulsion type MI instruments are advantageous for being cheap, having low friction errors, and having the ability to measure both AC and DC voltage.

- Attraction Type MI - Similar to the repulsion type MI, an electromagnet is used. The two ends of the coil represent the two probes or reading terminals for the voltmeter and are attached to a source. A current then flows through the coil and turns the coil into an electromagnet. A plate of iron is then placed in the magnetic field produced by the coil. Naturally, the iron plate will be attracted by the flux of the coil and move towards the coil a distance proportional to the strength of the magnetic field of the coil, which is also proportional to the amount of current flowing through the coil. Next, a pointer is attached to the iron plate and as the plate moves so does the pointer. A scale can then be fixed near the pointer to show voltage readings because the position of the pointer is proportional to the current in the circuit.

3) Electrodynamometer Type Voltmeter - Electrodynamometer type voltmeters were created because they have the same calibration for both AC and DC readings. This type of voltmeter is calibrated with DC, allowing us to measure AC without requiring additional calibration steps.

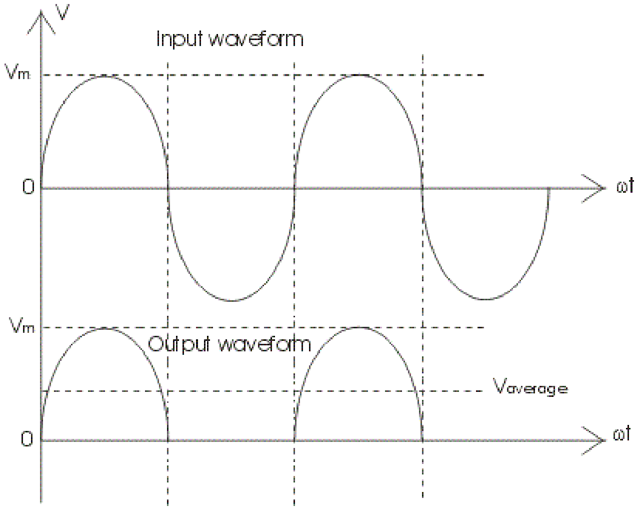
Let us focus on the structure of the electrodynamometer type voltmeter. This type of voltmeter consists of two coils, one moving coil and one fixed coil. This is similar to the idea in the repulsion type MI voltmeter, but coils are used instead of iron plates. The moving coil will be used to move the pointer with the help of a spring. A spring is used to minimize errors caused by the earth’s gravity. The current flowing through the moving coil will be limited to avoid heating. To do this, a very high resistor is placed in series with the moving coil. The spring will be placed on a pivoted spindle, allowing free movement of the pointer to indicate the correct value on a scale. This electrodynamometer type voltmeter has a uniform scale because the moving coil moves linearly over a range of 40 to 50 degrees. In this type of voltmeter, the moving coil acts and is referred to as a pressure coil. This is because the moving coil is connected across the voltage and the current flowing through the pressure coil is always proportional to the voltage.

As for the fixed coil, it is divided into two equal parts which are connected in series with the load. Consequently, the load current will flow through these coils. Dividing the fixed coil into two parts is done so that the voltmeter can carry a considerable amount of electric current and measure more values than just that in the micro- or milli-amp range. This gives the fixed coils the name of the current coils of electrodynamometer type voltmeters. Initially, these fixed coils were designed in order to carry current of approximately 100 amperes. However, today’s voltmeters are designed in order to carry current of approximately 20 amperes. This is done to save power and accommodate the need for mobile, compact, and long-lasting voltmeters.

Electrodynamometer type voltmeters have the advantage of using a uniform scale. However, the scale does contain a limit for measurement. This type of voltmeter is also a great choice because it can be used to measure both DC as well as AC voltages due to the fact that the scale is calibrated to measure record both values. Conversely, electrodynamometer type voltmeters have certain flaws. This instrument may suffer from errors in the pressure coil inductance, errors due to pressure coil capacitance, mutual inductance effects, errors resulting from improper pressure coil and current coil connections, errors due to Eddy currents, errors caused by vibration of a moving system, errors resulting from temperature changes or high temperature, and errors due to stray magnetic field. Magnetic fields outside of this electrodynamometer type voltmeter circuit can skew the results that the pointer and scale represent.

4) Rectifier Type Voltmeter - Rectifier type voltmeters, as their name suggests, implement a rectifier. A rectifier is basically a collection of diodes that allow current to pass only one way. Diodes provide zero resistance if it is forward biased (or if the current flows in the favored direction) and very high resistance if it is reverse biased. This important property of diodes is used to rectify voltages. The term rectify in electronics means converting an AC signal into a DC signal. This allows a rectifier type voltmeter to measure DC and AC voltages.

The rectifier type voltmeter is simply a rectifier used in conjunction with another voltmeter such as a permanent magnet moving coil voltmeter (PMMC). A rectifier makes up for the disadvantage of the PMMC not being able to measure AC voltage. When a DC source is connected to the rectifier type voltmeter, the diodes act as an open gate and allow the voltage to pass and be read by the voltmeter. As for the AC source case, the rectifier type voltmeter with a PMMC will also work and yield AC voltage readings. The diodes from the rectifier will limit the negative voltage or current flowing from the source, while allowing the forward biased half of the voltage or current to pass. As a result, the new voltmeter will not yield an average torque of zero for the entire AC clock cycle because only the positive half will pass through the rectifier and provide the voltage reading. The figure on the left shows AC voltage without a rectifier (top) versus AC voltage after it passes through a rectifier (bottom).



In addition, the rectifier limits the current drawn by the PMMC. This is important to achieve because if the current exceeds the current rating of the PMMC, it may damage it.

Rectifier type voltmeters are some of the most widely used instruments in the industrial world simply because of its cost. The cost of electrodynamometer type voltmeters is considerably high compared to rectifier type voltmeters, while the accuracy is basically the same. Furthermore, rectifier type voltmeters are not as delicate as others such as thermocouple instruments. Some additional advantages of rectifier type voltmeters include, having a frequency range of operation that can be extended to a high value, containing a uniform scale on the meter, and having low operating value of current and voltages.

5) Induction Type Voltmeter - An induction type voltmeter has two fluxes that are produced by two different alternating currents on a metallic disc. These two fluxes alternate and create an induced electromagnetic field. The electromagnetic field produced at one point then interacts with the alternating current of the other side to generate torque. The metallic disc levitates above the current coil due to Eddy current, and spins as a result of the electromagnetic field interacting with the alternating current. This spin is proportional to the amount of current and voltage of the load, and thus can be used in order to obtain a measurement reading of potential difference between two ports.

Induction type voltmeters have several advantages. They are inexpensive compared to moving iron type instruments, have high torque to weight ratio compared to other types of voltmeters, and they remain accurate over a wide range of temperatures and loads.

6) Electrostatic Type Voltmeter - Electrostatic type voltmeters use a static electrical field in order to produce a deflecting torque. These kinds of voltmeters are typically used in order to measure high voltages, but they can be used to measure lower voltages as well. There are two possible options of how the electrostatic force can act. First, two plates are oppositely charged in order to create an attractive force between them. One of these plates is fixed, while the other plate is free to move and is attached with a pointer. The free-to-move plate is attached to a spring to keep it in parallel with the first plate. Because of the attractive force between the plates, the free-to-move plate will move towards the stationary plate until the moving plate stores its maximum electrostatic energy. In the second option, there may be a force of attraction or repulsion (perhaps even both) between the two plates. When the two probes are connected to a desired element to be measured, the DC or AC voltage that we are interested in measuring generates a torque between the two plates. This repelling force rotates the movable plate, which in turn shifts the attached pointer along a calibrated scale. This is a very simple type of voltmeter.

Let us discuss the advantages and disadvantages of electrostatic type voltmeters. The first and most important advantage of electrostatic type voltmeters is their capability of measuring both DC and AC voltage. The reason for this is that deflecting torque is directly proportional to the square of the voltage. Second, the power consumption of this device is low because the current drawn by this type of instrument is very low. Another important property and advantage of this voltmeter is its ability to measure very high values of voltage. Electrostatic type voltmeters have a few disadvantages. This type of instrument is quite expensive compared to other kinds of voltmeters and they are also larger in size than its counterparts. As a result, these kinds of voltmeters are mainly used in companies that require to measure large voltage readings and do not have much domestic use because the average homeowner or user of voltmeters does not have a need to measure absurdly large voltages. Additional disadvantages of electrostatic type voltmeters include not having a uniform scale and containing various operating forces that are small in magnitude.

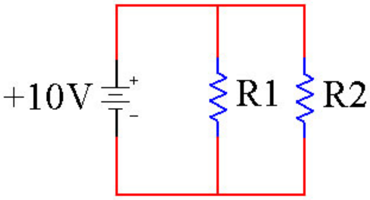
7) Digital Voltmeter (DVM) - The digital voltmeter is different from all the previous voltmeters discussed because it is the only voltmeter that obtains a measurement digitally rather than an analog reading. In other words, a digital multimeter directly indicates a measurement value rather than relying on deflection. Digital voltmeters are excellent instruments because they completely eliminate error due to parallax. Parallax is when a user attempts to read the pointer on a scale. With digital voltmeters, any two users will not have two results that differ by more than 0.01%. Similar to analog and digital clocks, digital voltmeters have a high-speed reading advantage. The user simply takes one

quick look and instantly reads the recorded measurement on the digital voltmeter rather than looking at a pointer and deciding which marker it is pointing to the closest. Additionally, high speed readings of digital voltmeters can be stored in memory for further analysis. This speeds up the recording process because none of the analog voltmeters have this capability. Digital voltmeters have other advantages such as being versatile and accurate, compact and cheap, having low power requirements, increased portability, and the ability to measure both AC and DC voltages.

For the purpose of this project, the team decided on choosing a digital voltmeter design rather than any of the above analog voltmeters. Voltmeter input impedance is the biggest difference between analog and digital voltmeters. Because most digital voltmeters have 50 times more impedance than analog voltmeters, digital meters are more accurate when measuring voltage in high resistance circuits. In section 3.3, we will discuss the significant impact that the input impedance of a voltmeter can have on the accuracy of your measurements. It is important to select a digital voltmeter for our design to minimize this “loading effect.” The loading effect a voltmeter has on a circuit is determined by the total resistance of the measured circuit in relation to the impedance of the voltmeter. The higher the input impedance of our voltmeter when compared to the circuit that we are desiring to measure, the more accurate our measurements will be. As a result, to maximize the accuracy of our readings, we are encouraged to build a digital voltmeter instead of an analog design. Another disadvantage of the analog voltmeters is that they are generally more bulky than the digital meters. This disadvantage results from the required long needle, scale, and coil that must be placed into an analog meter. Because the group is interested in creating a more portable and compact voltmeter, we again choose the digital voltmeter over analog due to this convenient and lightweight quality. Last but not least, one of the most significant perks of using a digital voltmeter over an analog version is the digital display. The display of a digital voltmeter is very easy to read because it shows one simple value after setting the voltmeter to the correct range. In an analog meter display, there is a scale with a pointer which shows the measured value. Not only is it difficult to discern between the many different markings on the scale, but the scale does not provide accurate voltage readings to several decimal points like the digital voltmeter does. In addition, analog meter displays are difficult to read because the user is required to close one eye and try to determine the most accurate marker that the pointer is aiming to. After considering the many differences between analog and digital voltmeters in regards to convenience, portability, and accuracy, we decided to proceed with a digital voltmeter design.

3.3 Voltmeter Measurement Technologies

In this section, we will discuss how a common digital voltmeter works. An ideal voltmeter is a two terminal device that measures the voltage (or potential difference) between the two terminals. Ideally, a voltmeter has an infinitely large resistance. As a result, no current would be lost from the measured circuit, and only the voltage will be computed because the two terminals would be in parallel. A fundamental rule of circuit theory is that any two elements in parallel have the same voltage. The design of the voltmeter abuses this law by connecting a very high resistance in parallel with a circuit element, thus showing the voltage recorded at that part of the circuit. Let us observe a simple example to better understand the process that voltmeters go through. On the left, we have a thevenin equivalent circuit with a Voltage of 10V and Resistance R_1 that can correspond to any circuit that we are interested in measuring. In this example, we are interested in measuring the voltage across resistor R_1. To do this, we connect our voltmeter in parallel with resistor R_1. Resistor R_2 symbolizes the high input impedance of our voltmeter. Initially, the voltage across R_1 is 10V and the current passing through R_1 is 0.01 Amps when R_1 is assumed to be 1kΩ. If the input resistance of our voltmeter is approximately 10MΩ, which is a fairly popular value for input impedance for a voltmeter, then we can calculate the expected current draw from the circuit by our voltmeter. Since our voltmeter is connected in parallel with resistor R_1 then our input impedance voltage should also equal 10V. Therefore, the current flowing into our voltmeter is approximately 10V/10MΩ, which then gives us 1µA (or one microamp). This value is 0.01% of the original current that flowed through R_1 prior to connecting our voltmeter. Therefore, connecting a voltmeter in parallel with a circuit element or multiple elements will draw negligible amounts of power and will have virtually no effect on the circuit, while at the same time providing an accurate reading of the potential difference that we are interested in measuring.



Similarly, the circuit can be analyzed and compared prior to and after the connection of the voltmeter to the circuit. When two resistors or impedances are connected in parallel, their total resistance or impedance can be calculated by the following formula:

$$R_{total} = \frac{R_1 . R_2}{R_1 + R_2}$$

In the example above, R_1 is equal to 1kΩ and R_2 is 10MΩ. Applying the above formula for R_{total} we obtain approximately 999.9Ω. Again, it is proven that connecting a voltmeter to a circuit leaves the circuit almost completely unaffected.

Voltmeters have the advantage over ammeters because they do not require the user to alter or break the circuit in order to obtain a measurement. The power loss in a voltmeter is V^2 / R_V where V is the voltage that needs to be measured and R_V is the resistance of the voltmeter. As a result, voltmeters should have a high electrical resistance in order to draw a very small amount of current and consequently consume a small amount of power from the circuit.

Analog inputs of a voltmeter usually have a limit of how much voltage can pass into the port. For example, an Arduino board has a default analog reference of 5V. As a result, Arduino can only measure voltages between 0V and 5V by default. These values can be increased by applying voltage divider and reducing the amount of voltage that is inputted to the Arduino analog port. Additionally, the reference voltage can be altered to give us more or less resolution. Let us observe a similar example but with an analog voltmeter that applies voltage divider to increase the range of measurement:

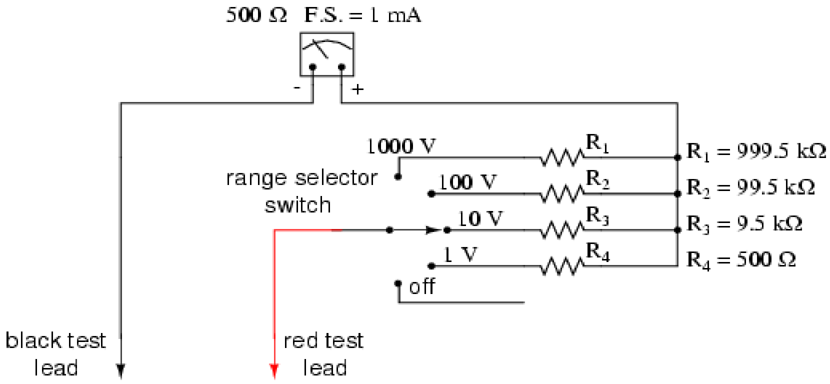


Figure 3.3: Selector switch (independent ranges).

In the diagram above, the voltmeter circuit contains a range selector switch from the positive probe. This range selector allows the user to choose which measurement scale he or she wants to apply. Like in the above case, the user can select between 1V, 10V, 100V, and 1000V ranges. The higher the range that the user wants to measure, the higher the resistance is required in order to limit the voltage seen by the analog input of the meter or analog to digital converter (Digital Voltmeter case). However, because resistor values such as 999.5kΩ, 99.5kΩ, 9.5kΩ, and 500Ω are difficult to find, more common resistors are used, and added in series such as the figure below:

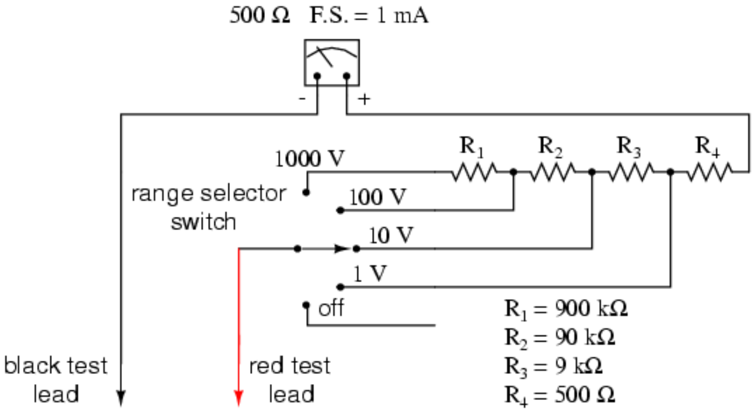


Figure 3.4: Selector switch (dependent ranges).

The difference between the two circuits is that in the prior circuit, only one resistor was used for each range. Conversely, in the latter circuit, multiple resistors could be used in series and the higher the range, the more resistors were used. This is simply done for convenience because more common resistors are implemented in the circuit. Resistors with values 900kΩ, 90kΩ, 9kΩ, and 500Ω are much more likely to be located in the average resistor bin. However, no noticeable difference will be seen from the user's perspective whether the user decides to implement the first circuit or the second circuit.

3.4 Safety Hazards and Protective Measures

Before selecting the safety measurements for our meter, we must identify all of the possible hazardous phenomenon that could occur when using our product. The most obvious threat are high voltage transients. According to Fluke's "ABCs of Multimeter Safety," voltage spikes can be generated from motors, capacitors, and power conversion equipment such as variable speed drives. Additionally, lightning strikes on substations, power plants, and most commonly transmission lines can also cause high voltage transients. These transients can even occur on low-voltage power circuits and can reach values as high as many thousands of volts. In a case like this, the voltage rating alone is not enough information to tell the user how well the measurement device was designed to not break from these high transient impulses. This is where the user depends on safety margins already built into the voltmeter. Examples of safety measures and components will be further discussed in this section.

Significant improvements in measurement devices such as the voltmeter were made in order to combat transients. For instance, the nominal bus voltage of electric commuter railroads was normally read at 600 Volts. Voltmeters rated at 1000 Volts were easily able to measure this nominal bus voltage. However, these meters would break after only a few minutes of taking measurements while the train was operating. After studying this mystery, a closer look revealed that the train starting and stopping generated 10,000 Volt spikes! As a result, voltmeter input protection needed to improve immediately.

The most dangerous and most injury-causing event from a transient is an arc blast. An arc blast is more severe than an electric shock and causes more electrical injuries every year. Arc blasts occur when transients ride on high-energy circuits because these circuits can deliver very large currents. If a transient causes an arc-over, the high current can sustain the arc and produce a plasma breakdown. A plasma breakdown is an electric explosion that results when the surrounding air becomes ionized and conductive. The way to combat this dangerous issue is making sure that the measurement device with the correct overvoltage category is used. Section 5.5.2 will discuss the four different measurement categories in greater detail.

One device used for overload protection in voltmeters is fuses. A fuse is a very thin wire that melts or vaporizes when current passing through it exceeds a particular fuse rating. Fuses are available with current ratings from 1/500 Amps to hundreds of amps. The thin wire can made from aluminum, tin-coated copper, or nickel. Most fuses in electronic equipment are cylindrical in shape and are glass or ceramic with a metal cap at each end. The current rating and voltage are written on one of these two metal end caps. Fuses are placed near the start of the circuit on a measurement device such as a voltmeter in order to monitor the current going into the meter. If the current entering the probes of the voltmeter is too high for the circuitry inside the meter to handle, the fuse will melt and prevent further damage to the device. Although fuses are not very necessary because the high input impedance of a voltmeter already limits most of the current that can pass into the meter, fuses serve as an extra protection against overcurrent. Overcurrent can cause electric components to overheat and cause a fire. It is beneficial to incorporate a fuse into the voltmeter design because if worst case scenario occurs, it is cheaper to replace a fuse rather than purchase a new meter. Adding a fuse into a voltmeter also helps count for the user making a mistake and connecting the meter’s probes where they should not be touching. If lightning were to strike on equipment with a fuse, the user may open the device and, upon examining the parts, see that only the fuse was damaged rather than the power supply or other components.

There are two most common sizes for fuses: 1¼ by ¼ inch, and 5 by 20 millimeters. The first size is mainly used for automobiles. Both sizes are used for electronic equipment, however the latter is more popular for electronics circuit design. Electronics tends to prioritize small sized components in order to be able to design small, low-power, and compact devices.

There are two basic types of fuses: fast acting and slow blow type. First, the fast acting fuse, like its name suggests, opens very quickly when its particular current rating is exceeded. This type of fuse is used for analog meters because their input impedance is not as high as that of digital meters. As a result, if too much current flows through an analog meter for even a very small amount of time, its circuitry components can be destroyed. Second, the slow blow fuse has a coiled construction inside the glass. Slow blow fuses are designed in such a way as to open only when there is a continued overload such as a short circuit. The coiled structure in the fuse helps stop the fuse from blowing on just a temporary current surge. Below is an image that shows a visual comparison between a good fuse, a slightly burnt fuse, and a damaged fuse.

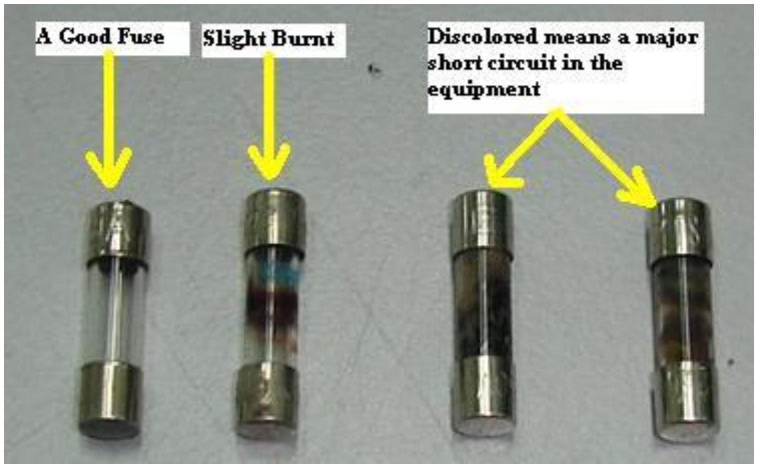


Figure 3.5: Good vs Bad Fuses.

Another method of testing a fuse instead of the visual inspection, is connecting an ohmmeter to both ends of the fuse. A good fuse will either show an ohm reading, or will read zero ohms. Contrarily, a blown fuse is technically an open. Therefore, a blown fuse will not show any reading on the meter.

A second possible preventative measure for input circuitry protection is varistors. Similar to how a fuse offers over-current protection, the varistor provides over-voltage protection. Varistors do this by utilizing voltage clamping, similar to a zener diode. That is, a varistor will trap extra voltage if the input voltage becomes too high, thus protecting the delicate circuit components of the voltmeter. The word varistor itself is an acronym, combining the words “variable” and “resistor.” This means that the resistance of a varistor is dependant on the amount of voltage flowing through it because the varistor changes its resistance value automatically with the change in voltage across it. In fact, a varistor is also known as a VDR, or voltage dependant nonlinear resistor. Varistors are made from semiconductor material and have symmetrical voltage and current characteristics, allowing them to be applicable to both AC and DC voltage implementation. The main function of varistors in input circuit protection is to combat transient surges. If lighting strikes near or onto a circuit which is being measured by a voltmeter, transients will travel through the circuit and can reach voltages up to several thousand volts. As a preventative measure, varistors are placed in parallel with the delicate circuitry inside of the voltmeter (a sample circuit can be seen to the left). By doing this, under normal measurement readings, the varistors act as high impedances and force most of that important current into the circuit of the meter. However, in the case of a transient surge, the high input voltage results in the varistors behaving as short circuits, thus forcing most of this current to travel to ground instead of damaging the significant and delicate electronic circuits and components.

There are two main types of varistors: the silicon carbide varistor (SiC), and the metal oxide varistor (MOV). Both types of varistors begin conducting current at a specific voltage and stop conduction when the voltage falls below a threshold voltage. However, the MOV has two significant advantages over the SiC varistor. First, the leakage current through the MOV's zinc oxide material is very small at normal operating conditions compared to the SiC varistor. In other words, metal oxide varistors allow the meter's circuitry to be more accurate and appear nearly nonexistent. The second advantage of metal oxide varistors is their speed of operation. MOV's clamp transients much faster than SiC varistors do. Metal oxide varistors are available in a wide range of varistor voltages ranging from approximately 10 volts to over 1,000 volts AC or DC. It is also important to note that the maximum continuous rms voltage rating of a varistor should be just above the highest expected supply voltage. For example, if the highest measurement that a voltmeter can record is 120 volts, then the rms voltage rating should be around 130 volts rms. Therefore, if an input voltage is greater than 130 volts, then the varistor will kick in and create a short to protect the circuitry of the meter from frying.

A third possible preventative measure for input circuitry protection is the zener diode. A zener diode also acts like a voltage clamp, similar to the varistor. However, a reverse biased Si small-signal diode breaks down at approximately 100 volts. This is much lower than a varistor. A zener diode is connected in reverse bias because the breakdown voltage of a zener diode in reverse bias is significantly higher than that of a zener diode connected in forward bias. Additionally, once a zener diode breaks down, it usually does so permanently. The zener diode breaks and acts as a short circuit with zero voltage across the diode. They can no longer return to their normal working state like varistors. Therefore, zener diodes may require additional maintenance compared to varistors. Another difference between a zener diode and a varistor is that a varistor is connected in parallel to an input voltage, while a zener diode connects in series to the input voltage. As a result, if a zener diode were to break down during the event of a transient surge, then the bulk of the transient voltage will go into the delicate circuitry and further damage the voltmeter.

A fourth possible preventative measure for input circuitry protection is a thermistor. A thermistor is a combination of the words “thermal” and “resistor.” Therefore, it can be deduced that a thermistor is simply a temperature sensitive resistor. Although all resistors are slightly affected by temperature, their temperature coefficient is quite minimal compared to the special high coefficient in thermistors. Thermistors can have negative temperature coefficients (NTC) or positive temperature coefficients (PTC). In NTC thermistors, the resistance of the thermistor decreases as the temperature increases. NTC type thermistors are generally used when a change in resistance over a wide temperature range is required. They are used more often for temperature sensors for the range -55 degrees celsius to 200 degrees celsius. NTC is most popular for having a quick response, being reliable, robust, and have a low price. Conversely, PTC thermistors experience an increase in resistance as temperature increases. PTC type thermistors are prefered when a sudden change in resistance at a certain temperature is required. Positive temperature coefficient thermistors have a type of temperature controlled switch that allows for its resistance to instantly increase. The most common switching temperatures for PTC thermistors are in the range of 60-120 degrees celsius. They are most commonly used in self-regulating heating elements and self-resetting overcurrent protection. PTC thermistors are a good choice for input protection in a voltmeter circuit design because it can help lower the input current if it reaches a dangerously high value. Thermistors are ceramic semiconductors and are most commonly composed of metal oxides.

A fifth possible preventative measure for input circuitry protection is the wirewound resistor. A wirewound resistor is a passive electrical component that limits current. It is basically a resistor that has a wire with a high resistivity wrapped around an insulating core in order to provide resistance. This is the oldest type of resistor that is still manufactured today. Wire wound resistor values can be very precise because its resistance value is dependent on the resistivity of the wire, the cross section, and the length, all of which can be accurately controlled by the designer. However, wire wound resistors can influence the current flow in an alternating current circuit, negatively impacting the results. This is because of the natural capacitance and inductance of the wire wound resistor. As a result, wire wound resistors are not ideal for high frequency operation within a circuit.

There are three types of wirewound resistors: precision, power, and potentiometer. In a precision wirewound resistor, the accuracy of resistance is very good, staying within plus or minus 0.05% of the design value for a particular circuit application. Power wirewound resistors are designed for very high power applications. The range of this type of resistor is between 0.5 watts until over 1000 watts. The function of power wirewound resistors as an input protection measure is to help dissipate heat. The third type of wire wound resistor is the potentiometer wirewound resistor. Wire wound resistors can function as potentiometers because of their durable construction. A potentiometer is basically a resistor with three terminals. Two terminals are the same as a regular resistor, but the third terminal is attached to a movable contact that varies the amount of resistance.

Wire wound resistors are often used in circuit breakers or as fuses. In order to function as fuses and breakers, a small spring is attached to one end of the resistor using solder to hold the spring in place. If the current passing through the resistor becomes too high, the solder will melt and the spring will pop up and open the circuit. This will serve as circuit protection for the vulnerable voltmeter components by stopping dangerously high current from passing through fragile components and damaging them.

PCB safety is another concern for this project. As a result, PCBs incorporate isolation slots to help separate nodes with high voltages. This is an additional safety measure to help account for arc-overs. Arc over is a case where voltage is so high that it passes through other even non-conductive materials and cause a short in the circuit. Arc-overs can also cause carbonization or burning on the

PCB and this would be undesirable because a permanent short would be formed. This would be irreversible damage and would require a new PCB to be rebuilt. Arc-overs can also occur over air, so there is an option to include a high-dielectric strength shield onto the PCB to counter this issue. Another advantage of PCB isolation slots is that dust cannot accumulate on the empty space. On a PCB with no isolation slots, dust can form on the board surface over time and reduce its dielectric strength.

Because no tool by itself can guarantee safety, there are several safety practices that the user should implement whenever he or she operates a voltmeter for maximum protection. First, the user should work on de-energized circuits whenever possible. However, if the user is not certain, then he or she should always assume that the circuit is live and dangerous. On live circuits, it is crucial to wear the correct protective gear. The user should use insulated tools, wear safety glasses or even a face shield, wear insulated gloves, remove watches or other jewelry, stand on an insulated mat, and wear flame resistant clothing instead of regular work clothes. Next, the user must take careful precautions when making measurements on live circuits. He or she should make it a habit to hook on the ground clip first, then make contact with the positive probe. When removing the probes, first disconnect the red or positive probe and then disconnect the ground probe last. For additional safety, the user can minimize his or her contact with the voltmeter by hanging or resting the meter on some object or surface instead of holding it in their hands. This will minimize personal exposure to the effects of transients. Another safety precaution is using the three-point test method to make certain that a circuit is dead or has no power. In the three-point test method, the user should first use the meter to test a known live circuit. Second, the user will test the target circuit. Third, the user will test the known live circuit again. This method will verify that the meter functioned correctly before and after the measurement. Last but not least, a little old electricians' trick of keeping one hand in your pocket can be used. This technique decreases the chance of a closed circuit to form across the chest and through the heart.

The last line of defense for users against transients or high voltages is the human body. The approximate body resistance under the skin from hand to hand across the body is 1000 Ω. At approximately 30 mA, respiratory paralysis occurs, thus causing a fatal situation for an average size human. Using Ohm's law, this means that only a 30 V voltage would be enough to cause death for a human. Fortunately, the body contains another powerful natural resistor, skin. The outer layer of dead skin cells protects the body and can help resist voltages up to 600 V from being fatal. However, this is assuming that the skin is not already punctured, meaning the user has no cuts and scrapes, and the skin is not covered in any conductive material such as water. A good voltmeter will eliminate any transient dangers before these transient currents ever make contact with the human. The safest meters contain double insulation, recessed input jacks and test leads with shrouded input connectors, non-slippery test leads with finger guards, and be made of high quality and durable non-conductive materials.

In the following section, we will discuss which safety components our team decided to include in the smart digital voltmeter design and the thought process behind why we chose these specific parts. Additionally, the standards section 4.1.1 will discuss the four different CAT ratings in greater detail and will expand on the uses of each category.

3.5 Strategic Components and Part Selections

In this section we will describe the various parts that we have researched and taken into consideration as part of our design. These include electrical components, software platforms, and packaging options. We will then discuss the pros and cons of each candidate component and come to a final conclusion on their use in our product. In section 3.6 we will give a brief summary on our parts selection.

When considering the various approaches to designing our meter, it is easy to see that there are a many different paths we can take. To begin our parts selection, it is necessary to define everything that the final product will entail. We are designing a digital meter that will measure a voltage when two conductive probes are connected together through the circuitry under inspection. This meter will read an AC or DC voltage, translate this voltage for user readability via some form of computation, and simultaneously display the value on a digital display while sending the information via wireless transmission to a smartphone application. In the application, the user will be able to read the measure voltage and adjust various settings such as sampling rates and data plot parameters. From this basic explanation of our product, it is easy to see our first area of interest needs to be how our voltage will be read and computed.

3.5.1 Processors and Operation Principles

There are several components that can be integrated into our product for the purpose of computing measurements and giving us the tools to create a functional meter. The first option we considered was using a microcontroller as the brain behind our project. A microcontroller, or MCU for short, is a small computer on a single integrated circuit. The MCU contains a Central Processing Unit, or CPU, some form of memory, and programmable input and output peripherals. As stated above, our product requires voltage to be read via wired probes into a component which can be programmed to compute these values and translate them into a readable display. With this definition, it is easy to see that a microcontroller would we a great option for our product.

Modern digital voltmeters have been optimized to run in the most efficient fashion using specifically designed processors to handle various operations. Most of these use what are known as chipsets. A chipset is a system of electronic components in an integrated circuit that handles the processing of data and flow of this data between the processor, memory, and peripherals. These integrated circuits are usually found on the motherboard of electronic platforms such as gaming consoles, home computers, appliances and many other hardware. They are designed to perform specific tasks in the most efficient manner. Companies such as Fluke design all of their multimeters to work with specific chipsets. These chipsets are developed by numerous companies and quite often designed per customer specifications. It is evident that this technology could be largely beneficial to our product and a viable option for our processing requirements.

In deciding between using a microcontroller or a chipset, we needed to consider the other design features of our product from our requirements specifications, along with the standards and constraints listed in section 4. The first and foremost constraint is the essentially the bread and butter of our product; the wireless transmission. There are multiple options for wireless transmission of data that will be further discussed in the next subsection (3.4.2). Before selecting which path to go down in that aspect, we had to consider which option, microcontroller or chipset, would be most compatible with wireless capabilities. Microcontroller platforms such as the Arduino, MSP430, and Raspberry Pi are largely known for their ability to easily interface with a multitude of peripherals and accessories. For example, both Arduino and MSP430 have shields which are produced and sold by various companies that have all of the components and hardware necessary for Bluetooth®, WiFi, NFC, and GSM wireless transmission. These shields are small PCBs that have been designed specifically for these applications and are relatively simple to incorporate with the microcontroller. Additionally, these microcontrollers have been around long enough that users have created various software libraries that make the coding process much simpler. This means that regardless of the path we choose for wireless transmission, using a microcontroller would allow our hardware and software design time to be very minimal.

For a pre-designed chipset, on the topic of wireless transmission, we realized immediately that we would have to design and build our own module. Regardless of the transmission type, we would need to purchase all of the hardware necessary for that type of transmission. We were quick to realize the impact this would have on our design time, cost, and the amount of extra space this hardware will take up on our PCB. Even though the cons heavily outweigh the pros in regards to wireless transmission, it is not an impossible task. This then led us to think of design time would be saved by using a chipset when considering the other hardware components such as input protection, voltage division, and display modules. As mentioned earlier, companies produce these chipsets specifically for voltmeter use which is the main selling point to companies like Fluke that produce voltmeters. This allows the company designing the meters to shorten their design time from 1-2 years down to 1-2 months! The companies manufacturing the chipsets have accompanying datasheets that have diagrams and demonstration board schematics for suggested use, with all of the components and their values already listed. If we were to use one of these chipsets, we could potentially cut our design time for the remaining circuitry by more than half. So even though we would need to spend more time designing the wireless capabilities, we would in turn save a lot of time on designing the voltage measurement and display components.

At this point in the selection process, the pros and cons of either choice were very evenly weighted. In regards to size constraints, both options were well within our requirement specifications. In terms of cost, both products with the required additional components were roughly the same, and well within our budget. We then did quick research and analysis comparisons on the power consumption and battery options based on data we could find in available datasheets for both options. For the various microcontrollers, all have generous power supply options, being that the platforms are designed with pre-installed voltage regulators. This means that we could power our microcontroller with voltages ranging from 1.8V all the way up to 20V, depending on the microcontroller. For us, this means that we can choose almost any power supply depending on our battery life specification to power our meter. On the other hand, chipsets are designed to work at very specific voltage levels, having various pins that require different voltage as is the case in most embedded system applications. This means that our design must be specific to whichever chipset we choose, and even though listed in the datasheets, requires many more components for reliable and accurate operation. In this sense, we determined microcontroller would be the best choice, but we still had more specifications to take into consideration.

The next important constraint we needed to consider was the electrical characteristics of each option and how they relate to our design specifications, mainly the accuracy of our meter in comparison to other products. Fluke is known as one of the highest quality multimeter producers in the world. Their products have been known to be as accurate as can be when it comes to digital measurement. We therefore chose to reference our meter with a Fluke 17b Digital Multimeter. Although this meter does more than voltage, the chipset used should perform all of the necessary operations for our product. The chipset this meter uses is a 4000 counts auto-range DMM IC, model FS9721_LP3 made by Fortune Semiconductor Corporation. This 4000 counts is the resolution of the ADC used in the chipset which can produce 3 and ¾ digits. While this accuracy level depends largely on the external electrical components, we can achieve this level of accuracy using the recommended demonstration design. If we were to use a microcontroller, we would be writing all of the code and designing the external hardware ourselves. This would mean our design needs to be optimal in order to achieve the accuracy levels chipsets like these produce. Fortunately, we can utilize additional components such as trim potentiometers to calibrate our meter within the required accuracy in the final design. The downside to this is the added PCB space and required user maintenance.

After considering all of the aforementioned options when choosing which processing path we will go down, it seemed to be neck and neck between the two possibilities. The last and arguably most crucial component to deciding between the two was total project implementation. Modern voltmeters are designed by experienced companies through thousands of man hours in all aspects of engineering, not just electrical. For our final product to be functional, we need to have a working system in which mechanical components and electrical components work together in unison. One of the chipset factors we have not yet considered are the mechanical requirements for the IC to operate correctly. The demonstration board schematics listed in all of the available chipset options show various switches in order to select different modes and user options such as range selection, hold functions, and other switching operations. If we were to use one of these chipsets and mirror our design on the recommended layouts, we would need to incorporate all of these switches and design our package to work with them properly. If we were to stray from the recommended use, we would need to figure out how to implement these changes without affecting the function of the IC. Regardless, the mechanical design of our product to use this chip effectively would account for a large amount of time dedication. Unfortunately, due to our largest project constriction which is time, we realized that we do not have the man hours nor the resources to complete this project on time using a chipset. On the other hand, microcontrollers are made for a wide range of user design options through several available I/O ports. This largely increases our switching options and design possibilities if our initial design ideas were not able to be realized. Even though using a chipset looked to be the more efficient option at first, mechanical switching implementation was the deciding factor for our group between the two options.

Once we decided that we would be using a microcontroller as the brain behind our meter, we had to consider the various options available and which devices could realize our product needs. The first microcontroller we considered using was the Arduino. Arduino itself is an open source electronics platform which allows users to create projects by making the pins of the main microcontroller IC easily configurable. This means that the input and output pins of the IC are routed on a single PCB to large ports in which users can quickly place and remove jumper wires without the need to solder. The pins on the IC used to control power, clock signals, and switching are taken care of through external components on the PCB. There are multiple versions of the Arduino with different parameters such as the microcontroller being used, operating/input voltages, clock speeds, memory, and input/output capabilities. Other microcontrollers such as the families of MSP430 and the Raspberry Pi have platforms similar to Arduino, differing mostly in the processors and the programming language. While we could do an extensive analysis on the various MSP430 and Raspberry Pi options, we did some quick research and decided to eliminate these options for a few reasons.

Our first concern with these platforms is our own familiarity. Due to time constraints, we had to consider what programming skills we already have in order to minimize design time. While we have some experience using the MSP430, none of us had ever used the Raspberry Pi. This microcontroller is programmed to run with Python IDE, rather than C/C++ languages used on the MSP340 and Arduino. The second factor in deciding between the MSP430 and

Arduino was available platform options. Arduino currently has 22 different platforms with several different processors, layouts, and many other characteristics to choose from. This means that we have several model options to sort through and select which platform would best suit our needs. Similarly, the MSP430 has over 500 microcontrollers available for purchase. In this aspect, regardless of which we choose, both brands have sufficient options to choose from in order to optimize our design. The final and deciding factor to take into consideration was available resources. While the MSP430 and Arduino have both developed user-friendly kits since around 2006, the Arduino has gained much more popularity due to the IDE and programming simplicity. Because the Arduino uses a very easy to use, high-level language and has numerous available libraries, users have found Arduino to be the first choice in personal project development. This translates to more resources for our group to access when we inevitably run into issues with our design. After taking careful consideration into all of the options, we decided as a group to use an Arduino microcontroller platform.

For our project, we are not allowed to use a development board that the Arduino platforms are built upon. Fortunately, Arduino gives a step-by-step procedure for building most, if not all of the external components that power and control the microcontroller. However, this procedure and build requires components that are specific to the ATmega328P microcontroller. Before jumping the gun and purchasing the required components, we must first review all of the additional components so that we know what specifically our microcontroller needs to do. If the ATmega328P is unable to meet our needs, we will need to consider other microcontroller options and thus design the supportive circuitry ourselves. In the following subsections, we will discuss the general design and layout of our meter, and then we will walk through the strategic selection process for the remaining components. Once we have a solid design, we will be able to determine if either of the microcontrollers can be used for our meter. Sections 3.4.2 through 3.4.5 will outline our strategic selection process for the external hardware that our final product will use. In section 3.5 we will review a few possible design architectures to help narrow our microcontroller selection which will be finalized and summarized section 3.6.

3.5.2 Wireless Transmission

Wireless communications began a new era in 1901, when M. G. Marconi successfully established a radio link in Morse Code between a land-based station and a tugboat (from Cornwall, England to St. John’s, Canada). Since this grand discovery, rapid improvements have been made in the field of wireless communication, allowing mankind to live in a new advanced digital age. In the beginning of digital wireless communication, its use was restricted to the military due to strategic requirements. Following Marconi’s wireless communication model, the military’s wireless systems consisted of a base station with a powerful transmitter that covered a certain geographical area. However, each of these base stations were independent of one another, not allowing the bases to communicate with each other. Conversely, today’s engineering advances in digital communications improved wireless communication worldwide. Not only are these base stations now connected and able to communicate with each other, but the cellular systems of today contain low-power transmitters. Low power resulted in low price, and low price allowed this technology to flow into the public and civilian hands.

Wireless communication serves to be one of technology’s greatest contributions to mankind. Wireless communication is the transmission of information over a distance without the need for wires, cables, or other electrical conductors. Transmission distance of wireless communication can range from a few meters (like in the case of a television’s remote control) to thousands of kilometers (like in the case of radio communication).

Wireless communication has several important advantages. First, wireless communication was improved to transfer information very quickly to consumers. Compared to wires, where the length of the wire creates a delay in signal transmission and reception. Second, the wireless capability allows users to access the Internet or specific devices anywhere and anytime without carrying cables or wires with them to wherever they go. In addition, this helps users save time by not having to travel to a certain location to transfer data. For instance, doctors and other professionals at their location of employment can all be in touch through wireless communication. A doctor can be instantly notified if he or she is required to perform an emergency surgery. Another advantage of wireless communication networks is that they are cheaper to install and maintain than wired networks. Wired networks require long wires to be buried deep underground, and a considerable amount of wiring is to be used in order to connect many buildings such as houses in a grid.

The main disadvantage of digital wireless communication networks is their easier susceptibility to hackers. It is very easy for hackers to intercept wireless signals that are spread through the air. Consequently, it is very important to secure private wireless networks so that shared information cannot be exploited by unauthorized users. Security measures such as WEP, WPA, and WPA2 are applied in many of today’s wireless networks.

WEP, or Wired Equivalent Privacy, is the most widely used Wi-Fi security algorithm in the world. It was even formally validated as a Wi-Fi security standard in September of 1999. WEP security initially provided 64-bit encryption and was later increased to 128-bit after US restrictions on the export of various cryptographic technology (64-bits) was lifted. Although 256-bit WEP encryption was later introduced, the 128-bit WEP encryption remained one of the most common implementations. However, even though the encryption algorithm increased its number of bits, it was not enough to keep up with the even faster increase in computer processor computation speeds. As computing power increased, it became easier and easier to exploit WEP encryption. To increase the public’s awareness of how weak WEP encryption was, in 2005 the FBI gave a public demonstration where they cracked WEP passwords in minutes all while using freely available software! Therefore, despite the attempts to improve the WEP system, it still remains very vulnerable and systems that rely on WEP are encouraged to be upgraded or replaced. The Wi-Fi Alliance officially retired WEP in 2004.

WPA, or Wi-Fi Protected Access, was the Wi-Fi Alliance's replacement to the now vulnerable WEP. It was formally adopted in 2003. The most common WPA configuration is WPA-PSK, or Pre-Shared Key. Compared to the 64-bit and 128-bit keys used in the WEP system, WPA used 256-bit keys. One significant difference with the WPA was implementing message integrity checks. These checks determined if a hacker had captured or altered packets passed between access point and client. Another important upgrade was the Temporal Key Integrity Protocol (TKIP). TKIP used a per-packet key system that was much more secure than the fixed key used in the WEP system. Advanced Encryption Standard (AES) later replaced TKIP. Although WPA was a direct upgrade to WEP, the TKIP of the WPA system was designed to be easily rolled out via firmware upgrades onto WEP-enabled devices. As a result, these WPA systems with TKIP were exploited because they reused certain elements from the WEP system.

WPA2 is simply called Wi-Fi Protected Access II. One of the most significant changes between WPA and WPA2 was the use of AES algorithms and CCMP, or Counter Cipher Mode with Block Chaining Message Authentication Code Protocol. This CCMP replaced TKIP, which was a weakness of the older WPA. The main security vulnerability of the WPA2 system requires the hacker to already have access to the secured Wi-Fi network. While connected to the network, the hacker gains access to certain keys and then attacks other devices on the same network. Therefore, home networks or other small networks are very safe with WPA2 security because it is easier to keep track of everyone that is connected to the network. Large businesses, on the other hand, be aware of this security weakness of WPA2. Wi-Fi Protected Setup (WPS) is used to facilitate the linking of devices to modern access points. This WPS is part of both WPA and WPA2, and remains as a vulnerability and potential access point for hackers. Although, it may take anywhere from two to fourteen hours of sustained effort with a modern computer for an attacker to breach a WPA or WPA2 security with WPS enabled. As a result, it is recommended for large businesses and companies with many different users accessing the same Wi-Fi to disable WPS so that virtually all major threats of hackers are removed.

The following are some possible types of short-range wireless communication:

1) Infrared (IR) Wireless Communication - This type of wireless communication applies infrared (IR) radiation. Infrared is electromagnetic energy at a wavelength higher than that of red light. Generally, IR wireless communication is used for shorter range communications as well as security control. In order for IR communication to work, the transmitter and receiver should both be located

nearby and in line of sight of one another. In other words, there should be no obstruction between the two. Infrared is often used in television remote controls and security systems. On the electromagnetic spectrum, infrared radiation is located between microwaves and visible light. As a result, infrared is able to be used as a source of communication.

2) Radio Frequency (RF) Wireless Communication - Radio frequency is a wireless electromagnetic signal that is mainly used as a form of communication in wireless electronics. Radio waves are a type of electromagnetic radiation with frequencies ranging from 3Hz to 300GHz. Frequency is basically the rate of oscillation of the radio waves, so the higher the frequency, the more cycles it undergoes per second. Radio frequency propagation is unique because it occurs at the speed of light and, as a result, they do not require a medium such as air in order to travel. Radio frequency waves even occur naturally from sun flares, lightning, and from stars in space. We humans communicate with artificially created radio waves that oscillate at various needed frequencies between 3Hz and 300GHz. Radio frequency communication is used in many of today’s industries such as television broadcasting, radar systems, computer networks, and remote control.

Radio frequency measurement can be divided into three major general categories: spectral analysis, vector analysis, and network analysis. First, spectrum analyzers provide basic measurement capabilities of radio frequencies. They are the most popular type of radio frequency instrument in many general-purpose applications. The main advantage of spectrum analysis is being able to view power vs frequency information. This will show what percentage of power is consumed at which frequencies. Spectrum analyzers also have the ability to demodulate analog formats via amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM).

The second category of radio frequency measurement is vector instruments. These are vector or real-time signal analyzers and generators. Vector instruments analyze and generate broadband waveforms. They are able to capture time, frequency, phase, and power information from a signal. These types of instruments are much more powerful than spectrum analyzers and provide excellent modulation control and signal analysis.

The third category of radio frequency measurement is network analyzers. Network analyzers are mainly used in order to make S-parameter measurements and other characterization measurements on radio frequencies or high-frequency components. Network analyzers are able to compare both the generation and analysis on multiple channels. However, network analyzers perform at a much higher price than spectrum analyzers and vector signal generators and analyzers.

Radio frequency instruments tend to operate in the very high frequency ranges. An added bonus resulting from high frequency ranges is immunity to some forms of noise and impairments as well as the size of the antenna that is required to receive such radio frequencies. The size of the antenna is usually ¼ the wavelength.

3) Wi-Fi Wireless Communication - WiFi stands for Wireless Fidelity and is synonymous with WLAN, or Wireless Local Area Network. WiFi, like most of the other wireless communications, is based off of radio frequencies. However, the radio frequencies of WiFi are completely different from walkie talkies, car radios, and cell phones because WiFi operates in the Gigahertz range of frequency, similar to microwaves. Car radios, for example, receive signals in Kilohertz and Megahertz frequency range, which are AM and FM stations respectively. More specifically, WiFi frequency is between 2.4GHz and 5GHz, while microwave operates at 2.45GHz. This is why some people may experience interference with their WiFi signal or in their internet connection when the microwave is cooking.

WiFi has many advantages. For example, the high frequency that WiFi operates at allows the signal to carry more data. This makes it an excellent choice for networks that transfer large files such as videos and online gaming. Data transmitted at 5GHz can move up to 54 megabits of data per second thanks to orthogonal frequency-division multiplexing (OFDM). This is a more efficient coding technique that splits a radio signal into several sub-signals before they reach a receiver. An added bonus of OFDM is that it greatly reduces interference.

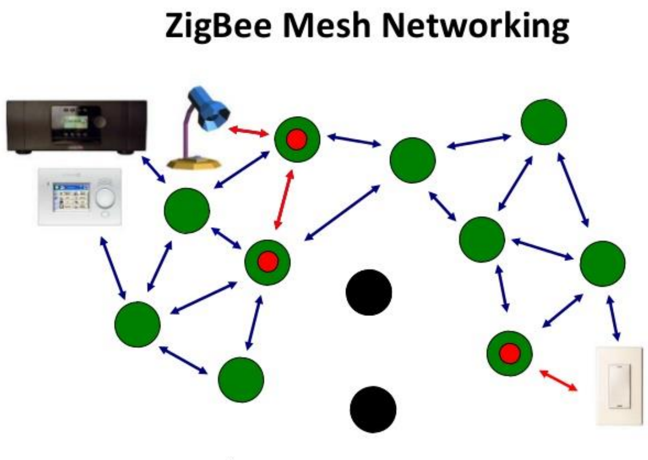
Wi-Fi wireless communication does have a disadvantage. Wi-Fi places all of its eggs into one basket because the networks relies on a strong router. In other words, it is also called a star network. If something were to happen to the router, then the entire network is disabled as a result. If the router fails, or if too many users connect to the router and use high-bandwidth applications at the same time, then the users can experience interference or lose their connections.

4) Bluetooth® Wireless Communication - Bluetooth® technology is designed to provide reliable, low-power, wireless communications over short distances. The original version of Bluetooth® used Gaussian frequency shift keying (GFSK) in order to transmit and receive data. GFSK is a form of modulation that is used in a variety of digital radio communications systems. It is able to carry digital modulation while still using the spectrum efficiently. This type of wireless communication is an improvement from other forms of phase shift keying because in the other forms the sidebands extend outwards from the main carrier, resulting in an interference to other radio communications systems in the same or nearby frequencies. Therefore, two different Bluetooth® devices could interfere with one another resulting in noise. In a later version, Bluetooth® introduced the Enhanced Data Rate (EDR) feature. This feature applied phase shift keying (PSK) modulation techniques, which allowed for data rates that were up to three times faster than GFSK. The main reason how multiple Bluetooth® devices avoid noise interference from each other (although they are in the same frequency band) is through a frequency-hopping technique called adaptive

frequency-hopping spread spectrum (AFH). Thanks to this special type of spectrum, Bluetooth® devices avoid interference, or noise, by quickly switching between 79 evenly spaced frequency channels from 2402 MHz to 2480 MHz. This hopping occurs at 1600 times every second, so any data lost due to interference is re-sent later over a different channel.

Although Bluetooth® is known mainly for its connections with headsets and microphones, Bluetooth® wireless communication can be an excellent choice for many other products. Benefits of Bluetooth® include, quick and easy wireless connectivity as well as lower power consumption and greater portability in comparison with with Wi-Fi.

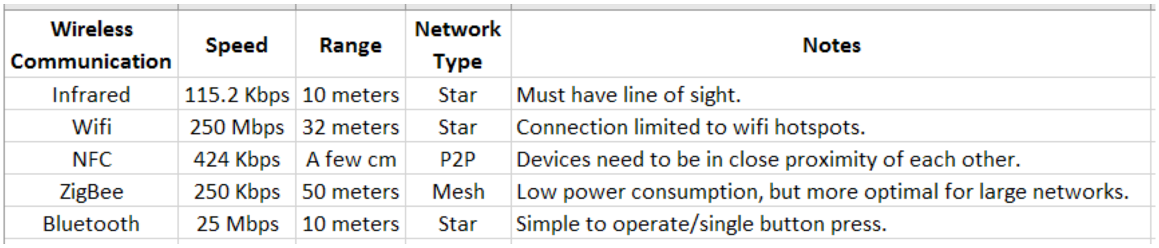
5) ZigBee Wireless Communication - ZigBee is a wireless language that allows everyday devices to connect to one another. The main advantage of ZigBee devices is that they can even transmit data over long distances by creating a mesh network of multiple ZigBee devices. Mesh network is unique from a tree network or a star network because those networks require to have a manager device or a priority device through which all data must pass. For example, Wi-Fi is a star network because all information must pass through a router in order to reach the computers connected to the Wi-Fi network. Therefore, if something were to happen to the router and disable it, the devices on the same Wi-Fi network would no longer be connected to the internet. By contrast, in a mesh network any of the devices on the network can be connected to one another and can pass information between and through devices. The figure to the left best illustrates how devices can be connected and transmit data in a mesh network. All circles symbolize a device in a mesh network, in addition to the lamp, the switch, and stereo system. Initially, the black circles were also devices in the mesh network, however, this illustration shows how ZigBee behaves under a situation in which two devices were damaged or ceased to work for some reason. Because of ZigBee unique mesh networking, the switch can take multiple paths to connect and communicate with the lamp, for instance. Before the two circles were damaged, the quickest path from switch to lamp led through one of the black circles. After these two black circles no longer functioned correctly, the mesh network provided an alternate path to retain communication between the switch and lamp.



ZigBee is mainly used for low data rate applications that need long battery life as well as a secure network. Additionally, ZigBee has a defined data transfer rate of 250 kbit/s, which is excellent and more than enough for data transmission from a sensor or an input device such as a voltmeter. ZigBee operates on 2.4 GHz frequency which is available for unlicensed use anywhere around the world. Consequently, a product developer can sell the same product using ZigBee anywhere on the planet. ZigBee is a popular choice for smart homes, connected lighting, and the utility industry. In smart homes, ZigBee provides home automation (controlling doors, windows, garage), home security systems, and appliances such as refrigerator, oven, microwave, etc. For connected lighting, ZigBee is used in residential as well as commercial lighting applications. ZigBee allows users to control LEDs, light bulbs, fixtures, remote controls, and switches. In the utility industry, ZigBee is used in products that monitor, control, inform, and automate the delivery and use of energy and water. Its main use is smart meters because they have a low data rate and use nearly no power to maintain. ZigBee controlled smart meters allow users to monitor their electricity and water usage.

6) Near Field Communication (NFC) - NFC is a short-range means of wireless communication that applies magnetic field induction to connect two devices. RFC is evolved from radio frequency identification (RFID) technology and operates as part of a wireless link. Communication between devices is allowed when they are touched together or brought within just a few centimeters of each other. NFC is a peer-to-peer (P2P) wireless communication system, meaning it only allows for two devices to connect and communicate by sharing data. To make up for the short distance constraint, NFC may be accompanied by communication technology such as Bluetooth® or Wi-Fi, which can be used for longer range communication and the transfer of larger amounts of data. There are several uses for NFC that will be covered. First, the user is able to use NFC to take pictures with a cell phone (with a built-in camera) and touch an enabled computer, television, or other cell phone to transmit the images for shared display. Second, the user can download applications or games to a handheld device or cell phone by touching a computer. Third, NFC can be combined with another longer range wireless technology to transfer large files between two devices such as a laptop and a desktop by bumping or touching the two together. NFC has no need for an internet connection, is easy to use, and works fast. Another significant advantage of NFC is that it does not require a pairing code to link up two devices because instead it uses chips that run on very low amounts of power. Therefore, it is much more power-efficient than other wireless communication types. A common example of NFC chips is credit cards. As for security, it is difficult for hackers to intercept data between two devices using NFC because the devices are so close together and the attacker will also be required to be located centimeters away from your device. The quick transfer speeds of NFC are also advantageous to protect against hackers, giving them a short window to intercept private data.

Now, each of these six short-range wireless communications will be compared to each other in an effort to select a wireless communication system that will best fit our design and recommendations for the smart digital voltmeter. The table below will summarize the key advantages and disadvantages of each wireless communication.



Wireless Communication	Speed	Range	Network Type	Notes
Infrared	115.2 Kbps	10 meters	Star	Must have line of sight.
Wifi	250 Mbps	32 meters	Star	Connection limited to wifi hotspots.
NFC	424 Kbps	A few cm	P2P	Devices need to be in close proximity of each other.
ZigBee	250 Kbps	50 meters	Mesh	Low power consumption, but more optimal for large networks.
Bluetooth	25 Mbps	10 meters	Star	Simple to operate/single button press.

Table 3.1: Short-Range Wireless Communication Comparison.

First, infrared wireless communication will be discussed. Although infrared is an effective means of short range communication, the dealbreaker was the constraint of infrared technology having to remain in line of sight of connected devices. This is a big concern for our digital voltmeter because we may want to constantly record data in order to create a graph of voltage readings versus time. This will be difficult with infrared technology because the recording device (in this case an android smartphone) is required to maintain a constant visual connection with the voltmeter and must remain in a line of sight. The user may accidentally drop or move their smartphone, resulting in a loss of connection and consequently will be required to restart the voltage recording from the beginning. Another issue with infrared technology is its inability to function through objects and walls. If the user is measuring and recording data from the digital voltmeter and attempts to pass this information to their smartphone, this connection will be lost if an object or someone were to walk through the line of sight link between the voltmeter and smartphone. Consequently, if a plot of voltage versus time is in the process of being recorded, data will be lost and the procedure will need to be restarted. Therefore, this type of short-range wireless communication is not suitable for our smart digital voltmeter design.

The next wireless communication in consideration will be Wifi. Wireless fidelity communication offers a considerable 32 meter range for connecting devices, which is over three times longer than most other wireless communications on our list. Wifi also provides the fastest data transfer rate available from our list of wireless communications, clocking in at approximately 250 megabits per second, compared to infrared data transfer of 115.2 kilobits per second. However, the main downfall of choosing Wifi for our smart digital voltmeter is the constraint of Wifi hotspots. In other words, our device would only work if we could connect a smartphone and the digital voltmeter to the same Wifi network. This would create a big problem in our design because then our instrument would only be able to transfer data to a smartphone if and only if both of these devices would be connected to the same Wifi network or hotspot. In addition, many Wifi networks are password protected, making it increasingly difficult to find an available hotspot to choose from in order to transfer data wirelessly through Wifi from the digital voltmeter to a smartphone. Another big disadvantage of selecting Wifi, is the length of time required to connect two devices via Wifi. Both devices would be required to input a password to connect to a password protected network and then establish a connection between the smartphone and voltmeter. Because our team wanted the smart digital voltmeter to operate at its full capability regardless of its location, the Wifi wireless communication option had to be rejected in favor of more convenient means of wireless data transfer.

The third wireless network that the group considered was Near Field Communication (NFC). Although establishing an NFC communication between two devices is very fast and easy, the main disadvantage of this type of connection is that the two devices need to be in close proximity of each other. This is a problem when it comes to connecting the digital voltmeter to a smartphone because we would need to keep the smartphone a few centimeters away from the voltmeter to maintain a stable connection between the two devices. Otherwise we may face similar problems to the infrared wireless communication. If the user shifts the smartphone slightly farther away from voltmeter, the wireless connection between the two devices may be disrupted. The connection would have two possible designs to combat this issue: 1) Continuously hold the smartphone steadily and very close to the voltmeter, and 2) Abandon NFC and adopt a wired connection such as USB. In option #1, this will cause great discomfort because generally both hands are already in use to hold the probes and attaching them to the desired terminals to be measured. This would be inconvenient for the user. In option #2, applying a USB cable would go against the entire project idea and design of the smart digital voltmeter with wireless capabilities. In conclusion, NFC wireless technology would not be an optimal candidate for the smart digital voltmeter.

The fourth wireless communication that the group considered was ZigBee. ZigBee shines when it is used to connect hundreds of devices together and still made affordable due to the chip’s much lower power consumption. This type of wireless communication specializes in connecting a very large number of devices together, such as street lights, and transfer the data between each device as well as all the combined data to a main device that the user can use to read all information. Normally the range of ZigBee devices is under 50 meters, but this range can be extended due to ZigBee special mesh networks, which allows for many different paths of data transfer between multiple devices on the same network. This is unlike Wifi networks where all devices must be connected to a master device (the wireless router) in order to function. Although ZigBee is an ideal means of communication for large networks, our smart digital voltmeter is a single device connected to a single smartphone. Therefore, ZigBee would not be used to its fullest potential if applied to our personal peer-to-peer (P2P) network. However, possibly in a future project, ZigBee can be used in order for a professor in a laboratory to view and grade students’ accuracy and results by connecting all voltmeters in the room together and conveniently observing their progress from the professor’s computer. Nevertheless, for this project, ZigBee is not the most optimal option for wireless communications.

Last but not least, is Bluetooth® wireless communication. This option seemed to have the least disadvantages when compared to how the team desired for the smart digital voltmeter to function. For our particular project, Bluetooth® is an excellent choice because of its versatility. Versatility, meaning that the Bluetooth® module in the digital voltmeter will allow for our device to connect wirelessly to any android smartphone in a very short amount of time. Setting up a Bluetooth® connection between two devices is so simple that it may even take only one simple push of a button and the user can begin utilizing the wireless functions of the smart voltmeter. In addition, Bluetooth® is also completely automatic, meaning that each Bluetooth®-enabled device can sense the presence of another device within a physical range. A common network protocol allows these devices to do this and share data. Bluetooth® devices can also remember other Bluetooth® devices in case they are separated out of each other's range. This allows the devices to recover their connection in a matter of seconds. Another advantage about this network protocol is that it reduces interference from other Bluetooth® devices that also exchange data nearby. The network protocol uses adaptive frequency-hopping spread spectrum (AFH). This allows Bluetooth® to limit noise interference between multiple devices because they quickly switch between 79 evenly spaced frequency channels from 2402 MHz to 2480 MHz. These switches happen 1600 times each second, so that any data lost due to interference is re-sent later over a different channel. One of the most important advantages of Bluetooth® is its compact quality. Bluetooth® chips are very small, and more importantly, their short range function only requires a tiny amount of radio power to work, which is less than a thousandth of a Watt. As a result, Bluetooth® is the ideal short range wireless communication for mobile battery operated devices such as smartphones and pocket voltmeters. Therefore, Bluetooth® is the best option for short range wireless data communication to use in conjunction with our digital voltmeter.

Because we will be using Arduino based circuitry/microcontroller, the group decided to go with the Solu JY-MCU HC-06 Slave Bluetooth® Serial Port Transceiver Baseboard Mini Module // Arduino Wireless Bluetooth® Transceiver Module Slave 4Pin Serial. This is a simple four-pin Bluetooth® module that is specifically meant to work with the microcontroller of our digital voltmeter. It has a relatively low cost for a Bluetooth® module at $7.39 and is small (4.1 x 2.8 x 0.2 inches) and lightweight (0.3 ounces), which is excellent for our project weight specifications.

3.5.3 Smartphone Operating System

The Smartphone market is saturated with a vast amount of options for developing, each with its own advantages which makes considering these viable operating systems a requirement in order to create the most effective and as widely usable as possible. These different operating systems vary drastically between resource allocation, development environment, data fragmentation, etc. User outreach is also very important when selecting an operating system to develop on, for instance if only 5 percent of potential application users prefer a certain operating system and that operating system alone is developed for, then 95 percent of potential users are dismissed or unable to even access the application, limiting use and outreach greatly.

As of the second quarter of 2016, Apple’s iOS has accounted for 12.9 percent of the smartphone using population in the world. This operating system is the second most popular operating system in the United States, largely behind Google’s Android at 86.2 percent. iOS development is generally done by Apple’s proprietary programming language known as *Swift.* Swift was released by Apple on October 2014 and is an adapted version of the existing language Objective C. Swift encompasses the key characteristics of Objective C while improving things like error handling and minimal instruction requirement, although Objective C still has its advantages such as faster compile times and wider library support. Apple allows either language to be implemented in their development tool, Xcode, making Objective C code and files able to be adapted to be recognized as swift code. It is important to not that Xcode is only available on Mac OS X, so developing for any Apple software through this Integrated Development Environment (IDE) will require the use of Apple’s own operating system.

Android is the most popular smartphone operating system in the world as of the second quarter of 2016. This means the outreach of the application will be widespread if the digital voltmeter were to be distributed on a drastically larger scale. The Android operating system uses the very common programming language known as Java for its application development making it rather versatile and user friendly if a user has developed and form of object oriented software in the past. Android development also takes advantage of XML files, which are used to easily manipulate graphical components of the application without needing extensive knowledge of the syntax. Android development can be done in multiple environments such as Eclipse (with the correct libraries added onto it), or Android Studio which includes a built in simulator to test Android applications. Android Studio has become one of the most popular IEDs available for Android development, and is available for multiple operating systems such as Windows, Linux, and even Mac OS X. One key advantage of Android versus most other smartphone operating systems is the Google Play Store. Google’s Play Store offers less restrictions and conditions in terms of applications allowable on the online store, so developing an application is not hindered by meeting strict online store policies, instead basic requirements and met and the application could potentially be available in a matter of hours after completion.

Although arguably the largest software company in today’s technological world, Microsoft Windows Phone has struggled to penetrate the market in terms of smartphone operating systems. Windows phone users account for about 0.6 percent of the world’s smartphone user population. Windows phone integrates Window’s personal computing operating systems such as Windows 10, Windows 8, etc. into the mobile operating system, connecting user data across multiple platforms. This would increase usability in terms of cross platform usage, however this adds to the intricacies of developing using the windows platform.

Microsoft windows phone can use many different languages in order to develop applications such as XAML for graphical interface creation and C#, C++, or Visual Basic for application behavioral code. Windows phone application development is usually done using Visual Studio, an IDE created by Microsoft and is available for Windows, Linux, and Mac OS X.

Linux has very recently entered the mobile operating system market with the introduction of Ubuntu Touch. This operating system is a mobile version of the very popular Linux distribution Ubuntu with some obvious adaptations to allow for use in mobile devices. This operating system may be among one of the most unpopular operating systems for mobile devices, as the users are generally required to adapt their current devices in order to use Ubuntu Touch, which requires a working knowledge of mobile phone software installation and special permissions. Following in the characteristics of its Linux based operating system, Ubuntu Touch is vastly customizable and open source, allowing and encouraging community develop in order to further improve the operating system. Development in Ubuntu Touch is very versatile, allowing QML or HTML5 to be used when developing the graphical interface, and JavaScript, Qt, Python, Go or some other programming languages to be used for the logical design of the application. The completed application is shared similar to that of Google's Play Store, however the outreach is fractional of the Android operating system. Ubuntu Touch is done by using the Ubuntu SDK IDE, which is only available for Linux Ubuntu 14.04 and newer.

RIM has many different branches in application development, whether an application is loaded onto a device with BlackBerry 10, BlackBerry OS, or a Playbook the development process may be slightly variant. RIM's Blackberry accounts for about 0.1 percent of all mobile operating systems in the world. Blackberry uses multiple SDKs based on the developing style the user prefers. Blackberry allows users to choose between many different languages of development also much like Ubuntu Touch. JavaScript, CSS, or HTML5 can be used for the graphical user interface while the logical design can be done using C, C++, ActionScript, AIR, Qt, etc. Even Java can be used to develop applications with a very similar development scheme to Android. BlackBerry allows development on many different IDEs as well such as FDT5 or Adobe Flash Builder 4.7 to list options.

3.5.4 Digital Display

Now that we have selected how our data will be wirelessly transmitted to the Android application, we will being looking that the other accessories that will turn our meter into a complete and total package. In this section we will review and select the digital display to be used in our final product. As in the previous sections, we will discuss the various options and the strategic selection process for each possibility.

There are several types of digital displays that can be used to display data. The first option we took into consideration was an LED display. This type of display uses seven light emitting segments to create one numerical digit. We can purchase various sizes that contain several digits and can be programmed to display a decimal point at any location. This type of display is common for counters, alarm clocks, timers and other applications for clear and readable values. We first considered this option because we believed it would draw low power, and we were not initially concerned having an advanced digital display being that the main use of our meter is to log data on a smartphone application. The second option we considered was an liquid crystal display, or LCD. These displays are used for televisions and computers, electronic products, and most importantly for us, they are the most commonly used display for digital meters. While there are other types of electrical displays, we had to quickly research their ability to interact with Arduino platforms. Fortunately, both LED and LCD displays have supportive software libraries that can be used to control all necessary functions, and because they are the most common types of displays, online resources for programming and controlling are readily available for our use. That being said, we quickly decided our product would use one of these two options.

LCDs operate on an electrical level by blocking light rather than emitting light as is the case for an LED display. This translates to a lower current necessary to power the device which, in turn, translates to less power consumption and a longer battery life for our product. LED displays must have a single digital I/O pin for each of the seven segments used to display a single digit. This means that in order to display a three digit value, over twenty output pins must be used! Fortunately, companies have developed specific integrated circuits for powering and controlling multiple-digit LED displays. One such example is the TM1637 LED Drive Control Special Circuit. This device uses CMOS technology which translates into very low current draw and power consumption. There are also accompanying Arduino libraries that can be downloaded for simple programming. Additionally, companies such as RobotDyn™ have developed products that use this driver in a small package to fully control a 4-digit LED display using only four pins. These four pins are power, ground, and two I/O pins to control data and a clock signal. This tells us that both LCD and LED displays are very comparable as it relates to power consumption for our product. Before selecting our display, we knew that our meter needed to display voltage measurements to at least the tenth decimal place. Luckily, using conversions within the code, we could display data for millivolts up to kilovolts on using the same number of digits. Knowing this, we determined that only four decimal places were necessary to display any range of voltage for our meter. Once we knew this, we could compare the size and cost options between the two options. After sufficient research, we found that both display options were nearly identical and well within our budget. We also found that the size of either options were within a few centimeters, and both compact enough to be used in our design.

Similarly to our selection process for the brain behind our meter, it seemed as if both options had an even amount of pros and cons up to this point. The next important factor to consider was the minimum number of I/O ports needed for each option. As mentioned before, the RobotDyn™ LED Drive Control Special Circuit needs only two digital I/O ports for full control. This is neglecting the power and ground connections which will be common for arduino and supportive circuitry because the regulated voltage output of the Arduino and common ground pins will be routed on the PCB to power all devices. An LCD, on the other hand requires several additional output pins to drive the display. Size options for LCD displays vary from as little as 8x2 (8 characters, 2 rows) up to 20x4 and larger. For our purposes, an 8x2 module would be sufficient to display the voltage being measured for any range. Even though this is one of the smallest packages, this device requires six digital I/O pins for full operation. Four pins control the data being written to the module, one pin enables the writing operation, and the last pin is used to select the rows and digits where characters will be displayed. It is necessary to realize that our final product must also control measurement range switching as well as AC and DC switching. The MCU must be able to sense these operations for accurate computation within the MCU's code. That being said, our final design must use additional dedicated I/O pins to let the MCU know which range and signal parameter (AC/DC) we are operating in. In section 3, we found that in order to measure voltage ranges from millivolts to around one kilovolt, we need at least four "R2" values in our voltage divider network. This means that we need at least five I/O pins for sensing (one for AC/DC selection). We knew that our design was subject to change and could require additional I/O ports, so we wanted at least two more in case we ran into issues. In total, our microcontroller would need 9 digital I/O ports for and LED display, or 13 for an LCD.

While the difference in necessary I/O ports seems to be large, even if we were to use an LCD, the ATmega328P has sufficient I/O pins, specifically 14, which would be enough for our product with two extra pins for wiggle room. That being said, we eventually realized that both of the displays would have nearly identical functional impacts on our final product. This led us to consider the final parameter to take into consideration which was how we wanted the to display our data. Initially, we believed a single value, the voltage being measured in real time, would be enough for our project. This is why we believed an LED display would be sufficient, because it would also use less power and we could select a smaller Arduino platform since it would use less I/O ports. But after our extensive research and analysis, we realized that we were not limited by either of these constraints and our display did not have to be limited to a single value. We could give our users more measurement values, such as average and minimum/maximum readings along with the real-time voltage being measured if we were to use an LCD. Additionally, we could now program the measurement units such as mVDC, mVAC, VDC, VAC, kVDC, kVAC, and any other text we might decide users would like. This was the determining factor for our display, and why we decided to use an LCD. It should be noted that because both of these displays are very cheap and well within our budget, we decided to purchase both in case we ran into trouble in the implementation of our project.

3.5.5 Diodes and Power Supply

Regardless of the microcontroller which we select for our final design, the external circuitry will be nearly the same. We have determined thus far that our platform needs at least 14 digital I/O ports for switching and digital display. We have also confirmed that we need to use both of the Rx and Tx ports for Bluetooth® communication. In following subsections we will determine any remaining ports necessary for our final design by discussing the remaining circuitry components and what our microcontroller needs to realize our final design. We will first discuss diodes as they will be crucial to AC measurement and power supply design. In section 3.4.6 we will discuss the input protection circuitry and necessary components followed by the remaining components that will be used for voltage measurement such as resistors, capacitors, and switches in section 3.4.7.

Diodes are two terminal components that allows the flow of current in one direction by making use of a PN junction. Ideally, this device has zero resistance in one direction and infinite resistance in the other direction. For our device, this technology will be necessary for various reasons. The first application of diodes we must consider are those used for AC measurement. As we have described, voltage measurement is done by using an analog-to-digital converter. This device only produces digital values for a positive input voltage. This means that for our device to measure AC voltage, we have to modify the signal before being input into the ADC. To do so, we can design a full-wave or half-wave rectifier, or modify the original signal without rectification. A half-wave rectifier removes the negative cycle of a sinusoidal signal by blocking current in the reverse direction with a single diode. Instead of removing the negative cycle, a full-wave rectifier inverts this voltage to a positive cycle using four diodes in a bridge. The common use for AC rectification is turning an AC voltage into a DC voltage. Therefore in either type of rectifier, a capacitor is connected between the output terminals to hold the voltage at a near constant level, only deviating by a small voltage swing. A full wave rectifier produces a more stable output with a smaller swing which is desirable for measurement accuracy. For our project, we designed three possible measurement options to consider which are listed below before selecting our diodes.

1) The first option would be to divide the AC voltage using the same network of resistors used for DC voltage division, and then pass the signal through a rectifier before being fed into the ADC. If we were to use a capacitor across the input of the ADC, we can turn the signal into a DC voltage and measure the peak values of the AC signal. Because the ADC would not see an sinusoidal signal, we would need to sense the frequency separately so that the data can be plotted on the smartphone application. This can be done using special pulse-width modulation ports on the Arduino, but could have undesirable effects on the voltage measurement circuitry. This method would work using either type of rectifier (half or full wave).

2) The second option would be to remove the capacitor and feed only the rectified wave into the ADC. Then, depending on our sampling rate, we could read all of the values of the positive cycle and generate a complete waveform from calculated averages. Unfortunately, our program would need to decipher where the minimum and maximum values in order to calculate the frequency. This could prove difficult with a fully rectified signal, but using a half-rectified signal, determining the positive sequence would be much easier by isolating the ADC values in between the strings of zeros produced by the rectified negative cycle. Therefore, this option would only need one diode.

3) The last option would be to divide the AC signal and add a positive DC offset. Because the voltage divider provides minimal deviation for each range, we know roughly how much voltage will be dropped on the ADC. We can then add half of this value as a DC offset so the bottom of the negative cycle would be at zero volts. The DC offset would be removed for voltage calculations and we would get the original waveform. Therefore, this option would not require a single diode for operation.

Because we want to use the least amount of components in our final design to minimize cost and size, we determined the third option would be the best choice for our design. As mentioned before, this option could possibly have undesirable effects which could hinder the accuracy of our meter. Therefore we decided to purchase enough diodes in the occurrence of this design not functioning properly. Because AC rectification has been a widely used technology for many years, companies have designed products to take care of these operation using only a single component. In the case that we must use option 1, we want the most accurate DC output, meaning the smallest possible ripples, and would need to use a full-wave rectifier. We therefore ordered a MULTICOMP W01 Bridge Rectifier Diode, which contains a full-wave rectifier on a single component. This component has a low maximum forward voltage drop of 1.1V and a forward current of 1.5A which is more than sufficient for our design. If we were to used option 2, we would need a single diode for half-wave rectification. We therefore purchased a 1N5818 Schottky diode. This diode is perfect for AC rectification as it also has low forward voltage drop and high current capacity.

The next possible application of diodes in our product are for the power supply. Before we can discuss this application, we need to once keep in consideration the possible microcontroller options. For the microcontrollers we are currently considering using in our design, all have an operating voltage around 3.3-5V. Because we will be implementing our own voltage regulation, we can design the rest of the circuitry based on our power supply selection. Once we have selected how our device will be powered, we can then discuss if diodes will be necessary for our power supply.

At this point in the selection process, taking into consideration the selected components to be used in our design, we have covered almost all constraints that would possibly prevent us from being able to use the ATmega328P microcontroller. Although we would ideally like to use the ATmega328P because of the available step-by-step guides for mirroring the development board along with PDIP package options which will aid in the breadboard design/testing, we still have the option to use other microcontrollers and design the supportive circuitry if need be. That being said, we will continue to examine and relate the remaining microcontroller possibilities before choosing a path. Our first area of interest for power supply was battery lifetime. We wanted a power supply that would be above the microcontroller's operating voltage has to offer a long life span, while also being an appropriate size for the meter. After taking in these considerations, we decided to use a 9V battery to power our device. This option would allow our product to operate properly until the battery discharged to around 6V for the worst case microcontroller option. We believe this option will be sufficient for our requirement specification of at least 200 hours.

Because we chose a 9V battery for our power supply, we had to consider the connector which would be used in our product. Our two options were a solid metal connector, or a wired connector. Because our final packaging design had not been complete at this point, we decided to purchase both connectors because of their cheap cost. Regardless of which connector we use, their common denominator is the ability to ensure the 9V battery cannot be inverted during installation due to their different sized positive and negative terminals.This brings us back to the topic of diodes. A common use of diodes for power supply is protection. This protection is for both the circuitry being powered and the power supply itself. If power supply polarity is accidentally inverted, this can potentially harm the microcontroller and supportive circuitry. Though it seems the 9V battery connector would prevent this from occurring, we may find ourselves needing to use a different power source down the road. In this case, we need to discuss how diodes can aid in the power supply and voltmeter protection by looking at a few different options.

The first and most common configuration placing a diode in series with the positive terminal of the power supply. If power supply, take a battery for example, is accidentally inverted, an inline diode will not be forward biased and no current will be passed. This causes the circuit to essentially be open and the power supply is protected from being short circuited. Unfortunately, when the battery is correctly installed, the diode is then forward biased and thus has a voltage drop, usually around 0.8V, which lowers the voltage being supplied to the circuitry. This is undesirable as it will shorten the battery's usable lifetime along with contributing to power loss. To help remedy this we can use a Schottky diode. This diode is designed to have a low forward voltage drop, around 0.4V depending on the specific diode through decreased resistance across the PN junction. While this is better than a traditional diode, we can do even better. Our third option is to use a P-channel MOSFET where the gate is connected to ground, the drain to the positive terminal of the battery, and the source to the top of the load. In this configuration, when the power supply is is connected properly, the transistor is actively biased, meaning the voltage between the gate and source (V_{GS}) is sufficiently negative, and thus current flows and power is delivered to the circuitry. When the polarity is inversed, V_{GS} is positive and current will not flow from drain to source and the circuit will essentially be open. Additionally, when the battery is inserted correctly, the resistance from drain to source ($R_{DS(on)}$) is very low, depending on the quality of the component. This will therefore contribute magnitudes less power loss while also protecting the circuitry and power supply. While we still plan on using a 9V battery which does not necessarily need this extra measure, we decided to purchase a 60V P-channel MOSFET made by Fairchild Semiconductors. This component has a maximum $R_{DS(on)}$ value of 70mΩ which is low enough so our 9V battery will not drop a large voltage and the power loss will be minimal. Additionally, in case we run into trouble with this arrangement, we decided to purchase a 1N5818 Schottky Rectifier diode. This diode has a very low forward voltage, around 0.4V with 250mA current draw.

Before moving on to the remainder of our electrical circuitry for safety and measurement, we will finish selecting the additional components needed for our power supply. At this point in the selection process, we have slowly narrowed down our microcontroller options to just two. These are the ATmega328P and the ATmega32U4. While we have determined that both options have sufficient I/O ports and also meet the rest of our requirements, there are a few differences that separate the two such as built in functions for programming and serial communication, as well as external power regulation requirements. We will save the other comparisons for later and focus now on the power supply. The ATmega328P requires a 5V regulated input, while the ATmega32U4 requires both a 5V and 3.3V regulated inputs. The latter option would require an additional voltage regulator along with two additional capacitors for operation. Although we would like to use the fewest components for cost and size concerns, we have yet to analyse the remaining differences as mentioned before and therefore we will discuss the power supply design and necessary components for both options.

For both cases, we need to have a 5V regulated voltage to the microcontroller's V_{CC} port. To do do this, we will be using a LM7805 voltage regulator from Texas Instruments. This product takes an unregulated input voltage from 7-25V and outputs a regulated voltage determined by the test conditions such as output current and power draw. From the LM7805 datasheet, for test conditions of V_I= 7-12V, I_O= 5mA to 1.5A and $P_D \leq$ 15W, the device will produce a regulated 5V output with a maximum swing of $\pm$ 50mV. The device requires two decoupling capacitors, one from the input pin to ground and one from the output pin to ground. These capacitors help to filter out high-frequency noise to produce a smoother output. From Arduino's tutorial on building your own development board using the ATmega328P, they use two 10μF These three components along with the 9V battery+connector and P-channel MOSFET described earlier are the only components necessary for our 5V power supply to the board. This setup will be used regardless of the microcontroller we select. If we are to use the ATmega32U4, we have to include two additional components for 3.3V regulation. From Arduino's website, we obtained the schematics for the Arduino Micro which uses the ATmega32U4. This platform has roughly the same 5V regulation as described above, with an additional LP2985 voltage regulator and

1µF capacitor. The output of the 5V regulator is connected to the input and ON/OFF pins of the LP2985 with the decoupling capacitor between the output pin and ground. Although we are leaning towards the Atmega328P, we will still keep these components in consideration and will determine if they are needed once we make our final MCU selection at the end of section 3.

3.5.6 Safety Components

Now that we have identified usage of diodes for both AC measurement and power supply protection, we will discuss the components used for input protection to the meter. These components are of utmost priority when designing the input to our meter, as they will be the barrier between dangerous electrical phenomenon and the rest of the meter. We will discuss how these components will protect both the product itself and more importantly the user.

Considering the various hazards outlined in section 3.4, we determined several possible components for our input protection. In the case of high voltage transients, we knew that we needed varistors that react very quickly to clamp the voltage and shunt the current before entering the measurement circuitry. Additionally, we knew that when these components were activated, there would be large amounts of energy flowing through the device. As the varistors cannot magically absorb this energy, we have to include additional components to handle this. For this reason, we purchased a 5W wirewound resistor that would absorb all of the transient energy produced from the varistors shunting the current back to the power source. To match the upper limit of measureable range of our meter, we purchased three varistors with clamping voltages at 340V. When placed in series between the positive and negative leads, if a transient voltage spike above 1020V occurs, the varistors will activate.

The final measure we had to consider was protecting the meter and components from the large amounts of current that would be induced from the varistors activating. As the meter would eventually heat up if the user was unaware that the varistors had activated, we needed a component to stop the flow of current and open the circuit. To do this, we have two options: fuses or thermistors. The fuse would allow current to flow in normal conditions but would open the circuit when a large current is passed through the component. A positive-temperature coefficient (PTC) thermistor on the other hand is a self-protecting component. When a large current flows through this device, the body will heat up and thus the resistance will increase drastically, eventually up to the mega ohm region and open the circuit. As the fuse would take longer to heat up and blow, we decided to use a thermistor for our design due to its quick activation. Though we plan to use this component, we also purchased a pack of high-rupture capacity (HRC) fuses. As outlined in section 3.4, if large amounts of energy are present when a fuse is blown, the explosion can produce high-velocity shrapnel and desecrate the meter and possibly harm the user. To counter this effect, the HRC fuses are filled with sand or other chemicals to contain a large blast without rupturing.

During the testing stages, we will implement both of these components and determine which is best for our design.

3.5.7 Electrical Measurement Components

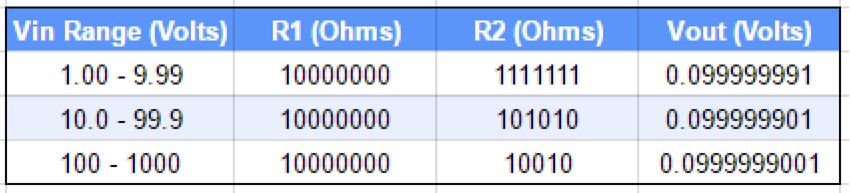
In this section we will describe how our microcontroller will be able to measure a wide range of AC and DC voltages and what components are necessary for this operation. We will begin by discussing the internal technology which will allow our microcontroller to measure voltage and then we will discuss the various design options which will allow this to happen.

Digital voltmeters operate on an electrical level by making use of an analog-to-digital converter, or ADC. An ADC is a system which converts a continuous-amplitude, continuous-time analog signal into a discrete-amplitude, discrete-time digital signal. There are several different types of ADCs such as the flash ADC, digital ramp ADC, successive approximation ADC, and many others. Microcontrollers such as the ones used in Arduino platforms have built-in analog comparators that serve many functions such as analog-to-digital conversion and multiplexing. For our product, we are interested in using the ADC to convert our input voltage into a digital signal which the microcontroller can decipher through code and generate the original voltage value that we can display and wirelessly transmit. Both the ATmega328P and ATmega32U4 use successive approximation ADCs. This type of ADC begins by using a comparator to compare the input signal to a predetermined reference voltage in sequence with a successive approximation register (SAR) and a digital-to-analog converter (DAC). Without getting into too much detail about the circuitry inside the system, the takeaway for our purpose is that the component needs only to have an analog voltage applied at the input for conversion. There is no current limiting requirements necessary for the system to function correctly. This is very beneficial in our case because the voltmeter must draw as little current as possible in order to make an accurate measurement and not interfere with the performance of the circuitry under inspection.

The ADC used in our two microcontroller options are both 10-bit and operate with input voltages from 0 to a specific reference voltage which is used for the internal comparators. This reference voltage (AV_{REF}) can be set by the user to various values, but the microcontroller automatically will automatically default to the operating voltage which we have established will be 5V. The resolution of the ADC is defined by the number of specific values it can produce. In our case, a 10-bit (N) ADC can produce 2^{10} or 1024 distinct values. The resolution is defined by the formula $AV_{REF}/(2^N-1)$. If we are to use the default 5V reference voltage, this gives us a resolution of 4.89mV. This means that when the input to the ADC changes by 4.89mV, the ADC is guaranteed to produce a new value. If we were to lower the reference voltage, let us say 1.1V for example, we can increase this resolution to 1.08mV. Unfortunately, this means that our range of input voltage is limited at 0 - 1.1V. Because our meter would be be useless if could only measure this range, we have to make use of additional components so that we can measure a large range without violating our input range of the ADC.

As we described in section 3.3, voltmeters must have a very high input resistance as to not draw current away from the circuitry under inspection. With this high input resistance in place and no current being drawn, the voltage at the ADC input will be the same as the voltage being measured. This means that we must use a voltage divider with a second resistor (R_2) and measure the voltage across this component. Using this method, we can select specific R_2 values that will allow for a small voltage drop within the ADC input range for a large range of input voltage. Even though using voltage division will increase our measureable range, we cannot use a single R_2 value for the entire range as lower input voltages will drop values on R_2 which are smaller than our resolution. Therefore, we must design a network of resistors that will be used for division for different ranges of measurement. This strategy was briefly defined in section 3.3. Before we can begin calculating these values, we must first identify the range of values in which our product should be able to measure.

Because our meter is intended to be used for electronic hobbyists, general household appliance maintenance and even electricians, we want our meter to be able to measure a steady-state voltage range of 1mV up to 1000V. As described in the previous section on safety design, this 1020V is the upper limit to what we can accurately measure as any higher voltage will be above the three varistors' clamping voltage and shunt the current before entering the IC. Although the safety components add some resistance to the input, we will select our nominal input resistance to be 10MΩ. This value will be the R_1 resistance for the voltage divider. We will calculate the remaining R_2 values by first addressing the ADC reference voltage as these must be calculated hand-in-hand. Both the microcontroller possibilities have the option to externally set AV_{REF} to any voltage. Because we want a high resolution, we will set this value to 1V. That being said, each resistor in our divider network must only drop a maximum of 1V onto the ADC. This gives us a resolution of 1/1023 or 0.9775mV. This resolution will ensure that when measuring in the millivolt range, our meter will be able to detect when the input changes by 1mV. It should be noted that when measuring in the millivolt range (1-999mV), the voltage can be directly fed into the ADC. Because our range grows larger for each resistor value, the precision of each range will decrease by 1mV, even though the optimal value would be our resolution of 0.9775mV. Using the voltage divider equation with the values of Vout = 0.1mV, R1 = 10MΩ, and Vin ranges being known, we calculated the R_2 values shown in Figure 3.1 below.



Vin Range (Volts)	R1 (Ohms)	R2 (Ohms)	Vout (Volts)
1.00 - 9.99	10000000	1111111	0.099999991
10.0 - 99.9	10000000	101010	0.099999901
100 - 1000	10000000	10010	0.0999999001

Table 3.2: Resistor Divider Network Values

When the input voltage for the first range (1.00-9.99V) changes by 0.01 volts, or 10mV, the output voltage being fed into the ADC changes by 0.0009999V or 0.9999mV. This change is slightly larger than our resolution value of 0.9775mV ensuring that we can measure down to 10mV of accuracy for this range. Similarly for the second range (10-100V), when the input voltage changes by 0.1V, we will see the same 0.9999mV change on the output meaning our meter can accurately measure down to 0.1V in this range. And as is evident, for the range of 100-1000V we can measure changes on the input down to a single volt. This design intuitively confirms that our product should be able to measure within 0.1% of the precision and accuracy that advanced modern meters are able to measure which meets our requirements specification.

Now that we know the resistance values we need for measurement, we can begin selecting our specific parts. Because these resistors must be very precise values, we determined that the tolerance must be as small as possible. We therefore decided to purchase 0.1% tolerance, through-hole resistors. Although we would ideally like to have surface mounted resistors for minimizing space, we had to acknowledge that we must test our designs on a breadboard.

As we mentioned previously, the R_2 values calculated above were based on our nominal R_1 value of 10MΩ which we knew would be affected by the added resistance of the safety components. Additionally, although we selected resistors with the best tolerance of 0.1%, we knew that these values had to be exact for our calculations to be correct. We therefore purchased a 13 value assortment of trim potentiometers that we will use to compensate for error in both the R_1 and R_2 resistors.

Now that we have determined how many resistors we need for accurate measurement, we can discuss how the range selecting will be implemented. To be able to measure off of each of these resistors, we have to have a separate analog input for each. Because we have four ranges, our microcontroller must have at least four analog inputs. Both the ATmega328P and the ATmega32U4 meet this requirement with the former having six channels and the latter having eight. While both can satisfy our requirement, it may prove valuable to have more channels as we may run into issues using the same channels for both DC and AC measurement. Though this can occur, we are still going on the assumption that we will be using the same network for both measurements. That being said, we must now consider the switching required to select the correct range.

After several different designs, we came to a final setup that would allow safe and efficient range switching for accurate measurements. The range selector will be a 4-bit, 2-position dip switch with all four switches connected together on one side as shown in Figure 3.2 below.

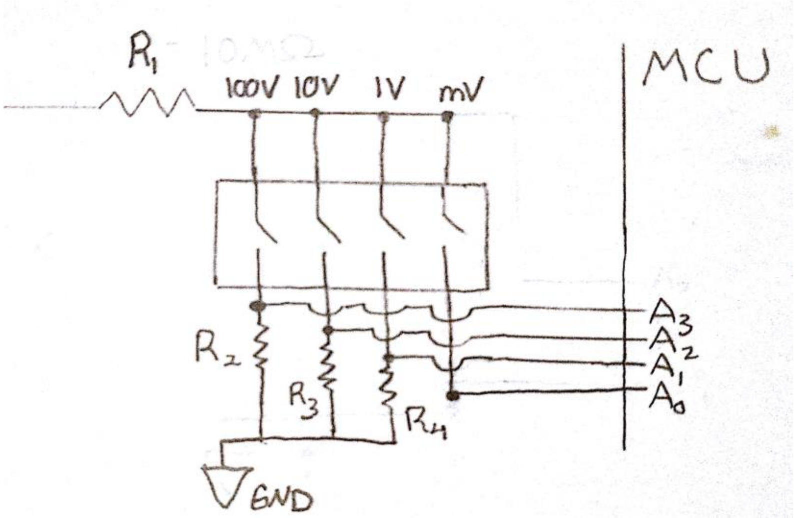


Figure 3.6: Range Selector Switch

In this configuration, the user can simply select the desired range of measurement by sliding a single switch. When no switch is selected, the circuit is open and no voltage will be measured. Additionally, when no range is selected there will be no connection to the ADC and the microcontroller will be protected. The most ominous design flaw in this configuration is the possibility of selecting more than one range at a time. This would effectively place two or more resistors in parallel, which would produce an inaccurate measurement. Fortunately, because we are using separate channels for each range, we can write code that checks if more than one analog input channel has voltage applied. We can then warn the user by illuminating an LED which would be placed near the range selector switch. Alternatively, we could connect a piezo speaker to sound a warning signal, or a combination of both LED and speaker. This part of the design will be determined after testing and further consideration. While there are many types of switches and methods we could use for range switching, we believed this to be the most effective design as it relates to cost, size, and functionality.

The final switch we will need for our design is for the power supply to connect to the microcontroller. Because this will only function to turn the meter on and off, we only need a two-position switch that will be placed in-line with the positive terminal of the battery and the P-channel MOSFET for reverse-polarity protection. We therefore ordered a 2P2T vertical slide switch from Microtivity. While we are on the topic of power supply, we also chose to add a single LED that will be powered by the 5V regulated supply in series with a 220Ω resistor to indicate when the device is powered on. This resistor is to limit the current flowing into the LED and therefore does not need to be a low tolerance component. In addition, this LED will not illuminate if the 9V battery is installed incorrectly.

The remaining resistors necessary for our project are needed for controlling the reset pin of the microcontroller, the brightness and contrast of the LCD, indication

LEDs, and for the AC/DC selection switch. For the microcontroller's reset pin, we must pull this to 5V (Vcc) in order to prevent the chip from resetting. In addition to this pull-up resistor, we will connect a small momentary switch to take advantage of the reset pin when writing new code and debugging. The pushbutton switch will pull the reset pin to ground causing the chip to reset. For this operation, we are going to use a 10kΩ resistor and a small momentary push-button that we have on-hand. For the indication LEDs that will be used for debugging the prototypes, we will be using 220Ω resistors to limit the current flow through the diodes. These indication LEDs will be connected to the Rx and Tx pins of the microcontroller to indicate when data is being wirelessly transmitted. Additionally, we will be using a permanent indication LED for warning the user when multiple range selection switches have been connected. If we choose to use a piezo speaker for warning instead, we will not need a resistor as this component operates through a pulse-width modulation (PWM) digital I/O pin alone. The last resistors that will be used in our product are the ones to control brightness and contrast of the LCD. We will use a 10kΩ potentiometer to determine the optimal levels and then use appropriate standard resistors to set these values. As they are not known at this point, we cannot determine their specific values until testing. As with the pushbutton for the reset pin, we will connect another switch with 10kΩ pull-up resistor that will be used to switch between AC and DC measurement modes.

Now that we have determined all of the necessary resistors for our project, we will now look at the capacitors necessary for remaining operation. As we mentioned before, we need two decoupling capacitors for the 7805 voltage regulator. We therefore purchased two 10µF ceramic capacitors rated at 20V which will be sufficient for our regulated 5V supply. If we are to use the ATmega32U4 which requires a 3.3V regulated power supply, we will also need a decoupling capacitor. We therefore purchased a single 1µF tantalum capacitor in the case that we use the ATmega32U4. The final capacitors that will be necessary for our microcontroller are the capacitors needed for the microcontroller's clock signal. Both of the microcontroller options operate using a 16MHz crystal oscillator and need capacitors to regulate the frequency of the crystal. The capacitance values needed depend on the load capacitance of the specific oscillator used. We purchased a COM-00536 16MHz Crystal from SparkFun which specifies the load capacitance to be 20pF. From Arduino's website, we obtained the open-source schematics for the Arduino Uno and the Arduino Micro which use the ATmega328P and ATmega32U4 respectively. Both of the schematics recommend using two 22pF capacitors, one from each pin of the 16MHz clock to ground. From the crystal's point of view, the two capacitors are in series giving an 11pF capacitance. The remainder of the load capacitance is taken care of from stray capacitance on the microcontroller's pin. Although the frequency of the crystal may not be exactly 16MHz, so long as the value is close to 16MHz, the IC will operate correctly.

3.5.8 Test Leads

A test lead is the combination of a conductive probe, cables and terminating connectors. There are several different types of test leads for circuit measurement such as voltage, current, and oscilloscope leads. Digital multimeters and voltmeters use simple voltage test leads with pointed probes, some having the option to attach alligator clips. Oscilloscopes on the other hand can use various complex and sophisticated leads for specific applications. Since we are designing a meter to only measure voltage, we need only simple voltage leads. We now must consider the various probe body options as they relate to safety and intended use. The probe body is composed of the conductive probe and insulated housing. For all options we will consider, the link between probe body and meter will be the same. The probe body will attach to a single conductive wire and then the male terminal connector which will mate with a female connector and subsequent traces on the PCB. The two most common probe types are pointed metal tips and alligator clips. Because the intended use of our meter is for quick measurements, we decided to use pointed metal tips as these are most maneuverable and inexpensive. With the probe type selected, we now had to consider the safety features and what will be necessary for our design. Because our meter will rated as Category III, our test leads must also conform to this rating. Though the meter ratings are based on ability to handle various levels of transient voltage spikes, the parameter that governs our test lead’s rating is the current handling capabilities. In the cases of high voltage transients, power supplies being shorted, component failures, or even user error, large amounts of current can be sent through the test leads. Although there will be numerous input protection components and procedures within the meter’s design, it is necessary to ensure our leads can handle large amounts of current without failure. Thin gauge wires will heat faster when current is passed through in comparison to thicker wires. A hot wire in-turn could cause the probe body to heat up enough, or even melt, possibly resulting in severe burns. We therefore need to ensure our wire which connects the probe to the PCB is thick enough to prevent this.

The second safety measure we have to consider while selecting our leads is the external design. Fortunately for us, Fluke has developed a safety guidelines document titled, “ABCs of Multimeter Safety.” In this document, Fluke lists four general safety guidelines when purchasing meters as they relate to the test leads. The first check is having double-insulated leads. The second is ensuring the meter has recessed input jacks and test leads with shrouded input connectors. The third check is to ensure the test leads’ probe body has non-slip finger guards so that the user does not slip and come into contact with the circuitry being measured. The last measure is that the meter and leads are made of high quality, durable, and non conductive materials. After considering this advice, we decided to purchase a pair of red and black, 1000V test leads with pointed probes manufactured by DMiotech. These leads have 4mm male banana jack connectors which will be complemented by two PCB mount, recessed female connectors. Although the leads themselves are not shrouded, the recessed female connectors will ensure that user skin cannot come into contact with the conductive metal on the meter.

3.6 Parts Selection Summary

Through large amounts of research and careful consideration of all aspects that could influence the accuracy and functionality of our product, we slowly narrowed down the possible processing technologies that could potentially be used as the brain of our meter. We first made the decision to use a microcontroller which led us to compare the various brands and models available. After considering the languages used and available resources for troubleshooting, we decided to use a microcontroller that can be programmed using the Arduino IDE. As there are several different MCUs used in the various Arduino platforms, we had to stop at this point and consider the rest of our design. After selecting all of the components that would be necessary to realize the remainder of the design, we narrowed our microcontroller options to two models: the ATmega328P and the ATmega32U4. Both of these options would meet the minimum design requirements of our project, so it came down to which product would be both cheaper to implement and easier to use on the breadboard. As the ATmega32U4 is a surface-mount device, we would have to solder individual pins to each of the contacts in order to build and test our designs on a breadboard. For this reason, along with the cost of the IC and supportive circuitry being cheaper, we chose to use the ATmega328P. This is a PDIP package type which can be directly placed on a PCB, and it can be mated with an adaptor socket for breadboard use. We purchased this socket along with the remaining circuit components that are needed to fully operate the device. These parts included an LM7805 voltage regulator, two 10μF decoupling capacitors, a 16MHz crystal oscillator with two 22pF capacitors, and a USB to serial Breakout board from SparkFun to program the chip with the Arduino IDE. Additionally, we needed to purchase a pushbutton that would be connected to the RESET pin of the MCU that would pull the pin to ground when we need to write new programs to the device.

After taking into consideration all of the possible remaining components for our final product, we chose several different components for each operation as we knew we could possibly run into issues with our potential designs. When researching the safety measures that would be used for our product, we came across several different technologies that could be used to protect the meter and the user from overload conditions such as high voltage transients. After extensive research for which technologies our product would need, we purchased the following safety components: 340V varistors, 365V varistors, a thermistor (PTC), fast-blow fuses, and a few 5W wirewound resistors. We also purchased Cat III rated test leads for measurement which will be connected to the PCB using female banana jack connectors. For the protection of the 9V power supply, we purchased a 60V P-channel MOSFET and a pack of 1N5818, as we were unsure of the final design at this point.

For the input voltage division, we needed several resistors and potentiometers to allow for a large range of measurement. We knew that our input impedance would have to be above 10MΩ in order to not compromise the the circuitry under inspection. For this reason, we purchased several of the following resistors to be used for the input measurement: 10MΩ, 9.5MΩ, 1MΩ, 100kΩ, 10kΩ, 1kΩ, 10Ω and 1Ω. Because these resistors must be very precise for accurate measurement and calculations, all of these components have a 1% tolerance. Additionally, we were aware that in order to adjust these values for maximum accuracy, we needed to be able to vary these resistors. We therefore purchased a set of trim potentiometers with values from 100Ω to 1MΩ. The resistors used in the remainder of the circuitry include: 10kΩ pull-ups, 220Ω resistors for the LEDs, 2kΩ and 4kΩ resistors for the Bluetooth® module, and the 100kΩ and 400kΩ resistors used to set the reference voltage of the ADC on the AREF pin.

For the digital display, we decided to purchase two different types as we may run into issues during implementation. The first device we purchased is a four-digit LED display with driver chip installed. After further design changes, we decided to purchase a 20x4 LCD which would be used instead of the LED display. Using this option, we could display more data to the users such as maximum, minimum, and average values when measuring in DC mode. In AC mode, we could display the peak voltage, the RMS voltage, and the frequency of the measured signal. To implement this display in our product, the only supportive component necessary is a resistor to control the contrast of the display. As we have not yet determined which value will be optimal, we will use a 10kΩ potentiometer until we decide which resistor value will be used.

For indication purposes, we purchased an assortment of various colored LEDs. We will use a single green LED that will be connected to the power supply to indicate when the device has been turned on. To indicate the user of which measurement mode, whether AC or DC, we will use two separate blue LEDs. One will be labeled as AC, and one as DC and will illuminate accordingly when the AC/DC selector button is pressed. The final LED is red and will be used to warn the user when more than one range selection switch has been activated.

In order to transmit the measurement data wirelessly to the smartphone application, we researched several options considering the pros and cons of each. After extensive research, we decided to use Bluetooth® transmission as this option will meet the minimum requirements and will be the easiest to implement. We will be using the Arduino IDE to program the microcontroller which contains specific libraries to aid in Bluetooth® transmission. We purchased a Solu JY-MCU HC-06 Slave Bluetooth® Serial Port Transceiver. To use this product, we only need to make the necessary connections to power, ground, and the RX and TX ports of the Arduino. However, as this module operates on 3.3V logic levels, we need to use a voltage divider with the two resistors mentioned above.

For the Smartphone application, after consideration of the varying operating systems discussed in the document, the application's developing operating system was finally decided to be designed using the Android operating system. The Android operating system has one of the easiest processes of application publication which is crucial for this project, as time is very limited. The operating system's application development process is also rather simplistic in terms of language knowledge as Java is one of the most popular programming languages and XML is a very visually based design language, with the GUI in Android studio allowing components to be placed at the visual level including size changes and position of these components on screen. Android is also the most popular operating system used worldwide, which allows the application to be distributed with the largest possible audience of users. Although this application is currently only under consideration of design using the Android operating system, the possibilities of expanding compatibility to other major operating systems is still present depending on the success and overall user satisfaction of the Android applications performance.

The remaining components needed for our product are the switches and LEDs. For the range selection, we purchased a 4-pin 2-position DIP switch. For the RESET pin of the microcontroller and the AC/DC selector, we purchased two momentary pushbuttons as these will only need to be pressed once for operation. The last switch is a 2-position slide switch connected between the battery and the input of the voltage regulator that will be used to turn the device on and off. We will be using a red LED to indicate when more than one range-selector switch has been enabled as a warning to the user of incorrect measurement. We will also be using a green LED that will be permanently connected to the power supply to indicate when the device has been turned on. The components that will be used in the final design are shown in Figure 8.

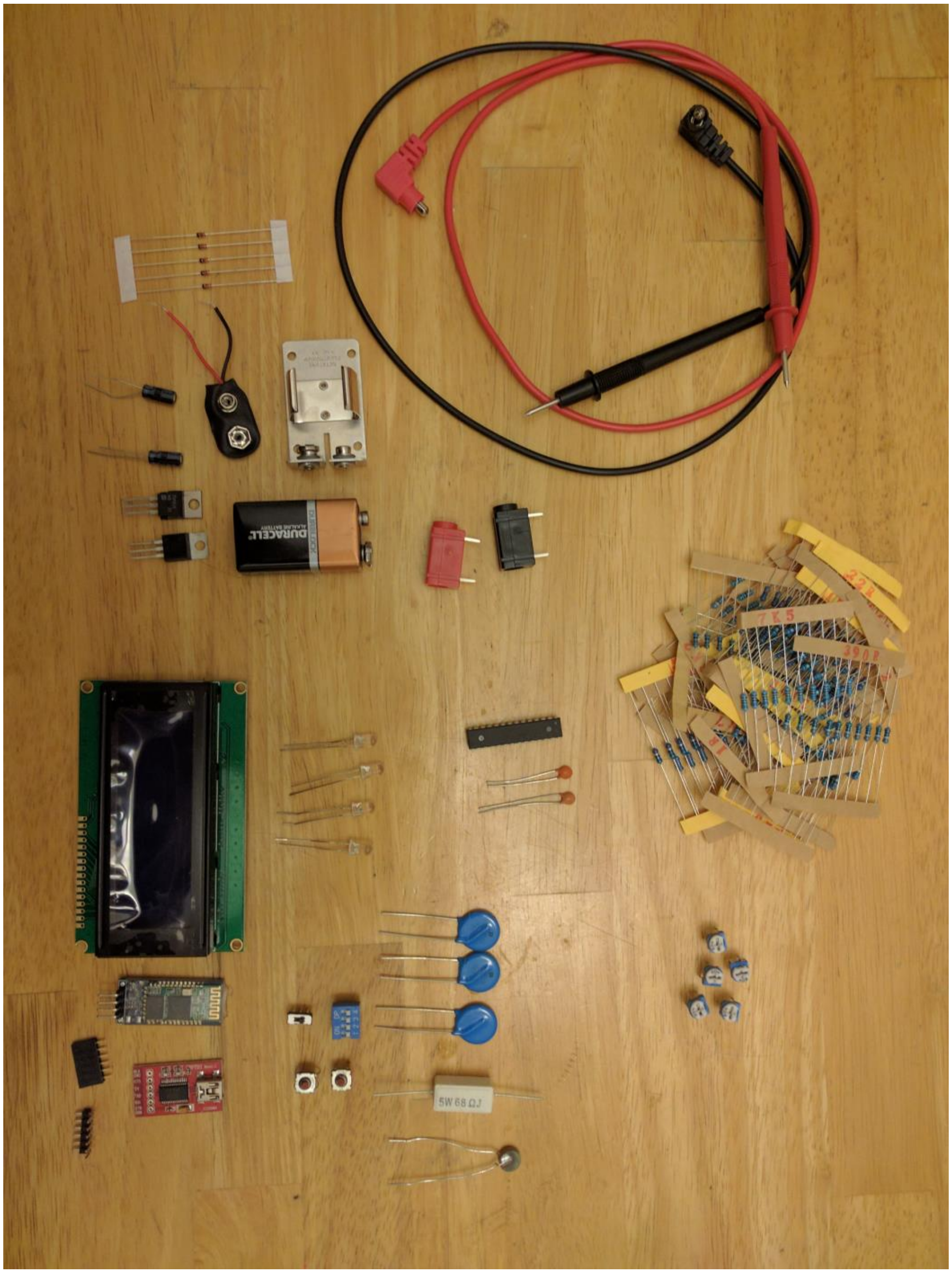


Figure 3.7: Parts

4 Standards and Realistic Design Constraints

4.1 Standards

In the following section, we will cover all of the standards that govern the voltmeter we are making. We will begin by discussing hardware standards. This is a very important section, because we must design our device according to the specifications of industry standards. These standards relate to the usage and safety of the voltmeter. making sure our design is made to within the constraints of the materials we are using is very important. A poorly design voltmeter or one that does not follow standards can harm users when they use it like giving an electrical shock that can cause in the human body being harmed (organs failing) and even cause death in some situations. Our voltmeter needs humans contact to make it work, this means that it needs to use the acceptable voltage that won't shock the user. Another area we need to focus on is looking at the materials that we be used in the voltmeter, we want to make sure the device's materials are not hazardous to the user and the environment. For each material we use and buy, we shall check to see if it is compliant with environmental usage. We will then discuss the standards necessary for software needed for our voltmeter. Basically they are two parts for our software, the first part is the software that will run on the microcontroller board we are designing. This software is closely related to the hardware's functionality, what this means is that the hardware mechanism must feed into the microcontroller and the software from there will take care of the data. the second part of the software is the application software that will run on the phone tablet. this software is mainly for receiving the data from the microcontroller and displaying to the user. A large part will spent on the hardware software constraints because we must make the hardware working at maximum capacity while locked within the standards. This part of the report will begin with the hardware standards, then proceed to the software standards. will work In section 4.2, we will discuss all of the possible design constraints that must be taken into consideration for our design.

4.1.1 Hardware Standards

Since we are designing a voltmeter with the expectations of selling it to consumers. It is very important that the voltmeter we are designing meets industry standards and is compliant with regulations and constraints of our market. the constraints are applied depending on what type of device we are making, different electrical appliances have different standards. Since we are making mobile electrical appliance, so it is normal that making our product around mobile standards of electrical components. For example our voltmeter will be mobile, so we need to design our system within standards of mobile multimeters. another hardware example is safety, the two are closely related, safety depends on the hardware used to make the voltmeter.

1. **Mobility Standards**

- There is not enough of information and official standardization on the mobility standards. However we did some simple deduction and came to the conclusion that our voltmeter must be as mobile as other devices of similar nature. The closest device to our voltmeter are multimeters, as matter of fact we can consider voltmeters as subsection of multimeters, being that the voltmeter is more specific (only measures voltage) than multimeters. The voltmeter is a subsection of multimeters therefore it is safe to say the standards of mobile digital multimeters also apply to the voltmeter that we are making. specifically we are designing a device that weighs the same as other industry multimeters and possibly less since it will have less functionality and components. The average weight is around 1 ounces. This is based off sampled data collected from official electronic seller online. we looked at the weights of ten multimeters which all have similar weights with littler (between 0 to 1 ounces) and averaged their weights. Current multimeters in the market is around **1 OZ based on the average taken**, we expect to make our product within the same range for the purpose of making it easy to carry around. Our voltmeter has the possibility of weighing less because it has less features on it (this could change).

Safety is very crucial when you dealing with voltages, the potential to be shocked is high and likely to occur, so we had to research what safety standards apply to voltmeters into safety standards as well. based on what we wanted in our system, our standard were divided into two parts. hardware and software, we shall begin with the hardware standards first followed by software standards later. we focused on these main standards:

2. **Safety Standards**

- ○ Our voltmeter shall follow the standards of the *"international Electronical Commission (IEC)".* We shall use the **IEC 61010-1 standard** because it offers a higher level of safety and the low-voltage (<1000V) requirement we will be using, it is also the test equipment needed for the highest levels of protection for our portable device. The hardware standard(s) that we will mainly be concerned with are the ***"measuring categories".*** The categories deal with voltage operation that the devices need to work on. We use what is known test instrument. It is rating system used to find the ability to withstand a voltage spike, which are applied through a specified level of resistance. The ratings are broken down into categories - CAT I, II, III, and IV. Before we talk about the different categories and which one applies to our voltmeter. It's important to talk about why we have such categories.
- ○ **Voltage Spikes (unavoidable hazards)** - multimeters are subject to much higher voltage than the user anticipated when taking measurements. There is a time when the user/ engineer has to measure a medium voltage appliance

in the range of 5000 V using a multimeter that can only measure 1000 V or less.

- Transients - are the most common voltages likely to happen and the ones that users should be protected against. hazards likely to occur because most likely users will have to measure voltage which are like to have high transient. These are some of the dangers of transients and which are likely to happen without warning.
    - Lighting Strikes - causes a transient on the power line, which in turn strikes an arc between the input terminals *inside the meter.*
    - High Fault Current - possibly several thousands of amps– ows in the short circuit just created. This happens in thousandths of a second.
- these are the categories.
    - CAT I: rated test instruments are signal-level tools for telecommunications and electronic equipment. Transient voltage risk is limited, but still exists, due to the distances between equipment locations and other equipment located between it and the primary electrical supply. The IEC no longer specifies protection levels for CAT I instruments. Under UL 3111-1, a Cat I 150V rated instrument must be protected to 800V. Under IEC 61010-1 2nd edition, a Cat I 150V meter could be protected only to 500V, as long as that information is in the user's manual.
    - CAT II: rated test instruments cover the local level of circuits for fixed or non-fixed power devices. This includes most lighting equipment, appliances, and 120V or 240V equipment inside a building.
    - CAT III: rated test instruments can withstand the transient voltage range found on most distribution circuits. These instruments are used primarily on fixed primary feeders or branch circuits. They're separated from CAT IV utility service or other high-voltage source equipment by at least one level or through transformer isolation.
    - CAT IV: rated test instruments are designed for testing on the primary supply source, which also includes 120V or 240V overhead or underground lines that power detached buildings or underground lines that power well pumps. The CAT IV rating covers the highest and most dangerous level of transient overvoltage electricians encounter when working on utility service equipment like exterior transformers.

Measurement Category	Working Voltage (DC or AC-rms to ground)	Peak Impulse Transient (20 repetitions)	Test Source Ohms = V/A
CAT I	600 V	2500 V	30 ohms source
CAT I	1000 V	4000 V	30 ohms source
CAT II	600 V	4000 V	12 ohms source
CAT II	1000 V	6000 V	12 ohms source
CAT III	600 V	6000 V	2 ohms source
CAT III	1000 V	8000 V	2 ohms source
CAT IV	600 V	8000 V	2 ohms source

Table 4.1: Transient test Values for measurement categories

- The table above breaks the voltages needed for each categories, It helps 'us' the engineers of the voltmeter to understand an instrument's true voltage "withstand rating". the Test procedures take into account **three main criteria: steady-state voltage, peak impose transient voltage and source impedance.** This table gives us an understanding of what is going on:

    - *Within* a category, a higher "working voltage" (steady- state voltage) is associated with a higher transient, as would be expected. For example, a CAT III-600 V meter is tested with 6000 V transients while a CAT III-1000 V meter is tested with 8000 V transients. So far, so good.
    - What is not as obvious is the difference between the 6000 V transient for CAT III- 600 V and the 6000 V transient for CAT II-1000 V. They are *not* the same. This is where the source impedance comes in. Ohm's Law (Amps = Volts/Ohms) tells us that the Ω test source for CAT III has *six times the current* of the 1 Ω test source for CAT II.
    - The CAT III-600 V meter clearly offers superior transient protection compared to the CAT II- 1000 V meter, even though its so-called "voltage rating" could be perceived as being lower. **It is the combination of the steady-state voltage (called the working voltage), and the category that determines the total voltage withstand rating of the test instrument, including the all-important transient voltage withstand rating.**

- **Creepage and clearance -** in addition to being tested to an actual overvoltage transient value, multimeters are required by IEC 1010 to have minimum "creepage" and "clearance" distances between internal components and circuit nodes. Creepage measures distance across a surface. Clearance measures distances through the air. The higher the category and working voltage level, the greater the internal spacing requirements. One of the main differences between the old IEC 348 and IEC 1010 is the increased spacing requirements in the latter.

- As you shown above safety category was setup to protect the user against transients, safety must be build into the test equipment. Our Task is to find which category our voltmeter fits in. The real issue for multimeter circuit protect is not just the maximum steady state voltage range, but the combination of of **both steady state and transient overvoltage withstand capability**. When transients ride on high-energy circuits, they tend to be more dangerous because these circuits can deliver large currents. If a transient causes an arc-over, the high current can sustain the arc, producing a plasma breakdown or explosion, which occurs when the surrounding air becomes ionized and conductive. The result is an arc blast, a disastrous event which causes more electrical injuries every year than the better known hazard of electric shock.
- For our voltmeter we are specifically going to use **“CAT 3”** because our device will not be directly connected to mains (AC power source). We will be using a portable power source likely a battery. consequently this come with benefits, having voltmeter not connected to “mains” allows the device to be protected, by making sure the it is connecting to a source circuit which measures limit transient: another benefit is the “overvoltage” or “high voltage” is calibrated to the right level, so the user doesn't have to worry about getting an electric shock during the voltmeter operation.
- **Overload Protection** - is another factor that has to be considered on top of CAT III, it is simply a protection circuit that clamps high voltage to an acceptable level. another addition is the the thermal protection circuit that detects an overvoltage condition, it protects the meter until the condition is removed and then automatically returns to normal operation the most common benefit is to protect the voltmeter from overloads when it is in ohms mode. In this way, overload protection with automatic recovery is provided for all measurement functions long as the leads are in the voltage input terminals.
- In conclusion to “safety standards section”, it was important for us to imagine the worst possible scenario where our voltmeter was going to be used, we looked at the demographic and the environment it was going to be used in. Once we finished that, we looked at the safety standards that applied to our problem. Once we finishing making the voltmeter, ***we are going to test it using the category we have chosen to see if it has met those standards and passed, if it is passes it will be certified as safe and compliant with regulation.***

3. Serial Communication standards

- In the world of embedded electronics and microcontroller, devices communicate to each other through communication protocols, there are many protocols (in the hundreds). but they can be grouped into two categories:

serial and parallel communication (NOTE: these communication use wired cables to enable communication between two or more devices).

- Parallel Communication
    - parallel protocols enables the transfer of multiple bits at the same time through buses of data (8,16, or more wires). the main advantage of this protocol is how fast it is. compared to serial communication.
- Serial Communication
    - This is the most common form of communication between electronic devices. Communication serially involves sending series of digital pulse back and forth between devices at a mutually agreed-upon rate. The sender sends pulse representing the data to be sent at the agreed data rate (bits per second) and the receiver listens for the pulses at that same rate. The devices both have clocks that determine at which rate they will exchange information.This protocol transfers data one bit at a time. they can operate on a single wire as well. what happens during serial communication is that 1) a common ground connection is set so that both devices have a common reference point to measure voltage by. 2) one wire for the sender to send data receiver on which is the transmitter line for the sender 3) one wire for the receiver to send data to the sender on the receiver line for the sender. For example suppose we have an exchange data at rate of 9600 bps and we would like to transmit it to another device the receiver will continually read the voltage that the sender is putting out, and every “1/9600 th” of a second, it will interpret that voltage as a new bit of data. If the voltage is high (+5V) it will interpret that bit of data as a 1. If it is low (0V), it will interpret that bit of data as a 0. By interpreting several bits of data over time, the receiver can get a detailed message from the sender. at 9600 baud, for example, 1200 bytes of data can be exchanged in one second. This is how serial communication works, while it send a lot of data it is not practical for our needs.
- Asynchronous communication
    - In this protocol data is transferred without support of an external clock signal like synchronous. It is good because it minimizes the required wires and I/O pins but it comes at a cost. you must make sure the data is transferred correctly or reaches the other end. **one of the most common types of Asynchronous communication system is Bluetooth®.** the benefits of using this protocol is that it does not require synchronization of both communication devices. It is also cheap because this type of communication requires less hardware. From now on we shall focus on Bluetooth® protocol.

4. Bluetooth® Protocol Standards

- **Bluetooth®** is a technology that describes communication between devices **using short range radio frequency(RF) technology that operates at 2.4GHz.** This technology is capable of transmitting voice and data at low power. This is why we are using it in our project, because we want to transmit the readings of the voltmeter to an Android application wirelessly. Bluetooth® is one of the technologies that enables this process (it is the de facto standard in the electrical and computing industry). This technology is safe, trustworthy and very secure due to several layers of data encryption and user authentication used. This is another appealing factor because it means our data cannot be interfered with. We will be using Bluetooth® to wirelessly synchronize and transfer data between the microcontroller attached to the voltmeter board and the Android application.

- **Bluetooth® Profile:** profiles are additional protocols that define how Bluetooth® works. These are additional protocols built on top of the basic available ones in the Bluetooth® standard specifications. Different devices have different profiles. in order for two devices to communicate they must have the same profiles otherwise they won't work. the specification are concerned with the use of Bluetooth® technology to support various applications. Each profiles specification is mainly concerned with the technology defined in the core specifications i which includes description of which aspects of the core specifications are mandatory, optional and not applicable. the main purpose of profile specification is to define a standard of interoperability, so the different vendors who sell Bluetooth® enabled components. that claim to support a given usage model will work together.
- Bluetooth® Protocol Architecture - Bluetooth® is defined as a layered protocol architecture consisting of core protocols, cables replacement and telephony control and adopted protocols. It consists of five - layer stack consisting of the following elements:
    - **Radio** - Specifies details of the air interface, including frequency, the use of frequency hopping, modulation scheme, and transmit power.
    - **Baseband** - Concerned with connection establishment within a piconet, addressing, packet format, timing, and power control.
    - **Link manager protocol (LMP)** - Responsible for link setup between Bluetooth® devices and ongoing link management. This includes security aspects such as authentication and encryption, plus the control and negotiation of baseband packet sizes.
    - **Logical link control and adaptation protocol (L2CAP)** - Adapts upper-layer protocols to the baseband layer. L2CAP provides both connectionless and connection-oriented services.
    - **Service discovery protocol (SDP**) - Device information, services, and the characteristics of the services can be queried to enable the establishment of a connection between two or more Bluetooth® devices.
    - RFCOMM is the *cable replacement protocol* included in the Bluetooth® specification. RFCOMM presents a virtual serial port that is designed to make replacement of cable technologies as transparent as possible. Serial ports are one of the most common types of communications interfaces used with computing and communications devices. Hence, RFCOMM enables the replacement of serial port cables with the minimum of modification of existing devices.

- Bluetooth® specifies a **telephony control protocol**. TCS BIN (telephony control specification—binary) is a bit-oriented protocol that defines the call control signaling for the establishment of speech and data calls between Bluetooth® devices.
- The **adopted protocols** are defined in specifications issued by other standards-making organizations and incorporated into the overall Bluetooth® architecture. The Bluetooth® strategy is to invent only necessary protocols and use existing standards whenever possible. These are the adopted protocols:
  - PPP - The point-to-point protocol is an Internet standard protocol for transporting IP datagrams over a point-to-point link.
  - TCP/UDP/IP - These are the foundation protocols of the TCP/IP protocol suite.
  - OBEX - The object exchange protocol is a session-level protocol developed by the Infrared Data Association (IrDA) for the exchange of objects. OBEX provides functionality similar to that of HTTP, but in a simpler fashion.
- Usage Models -

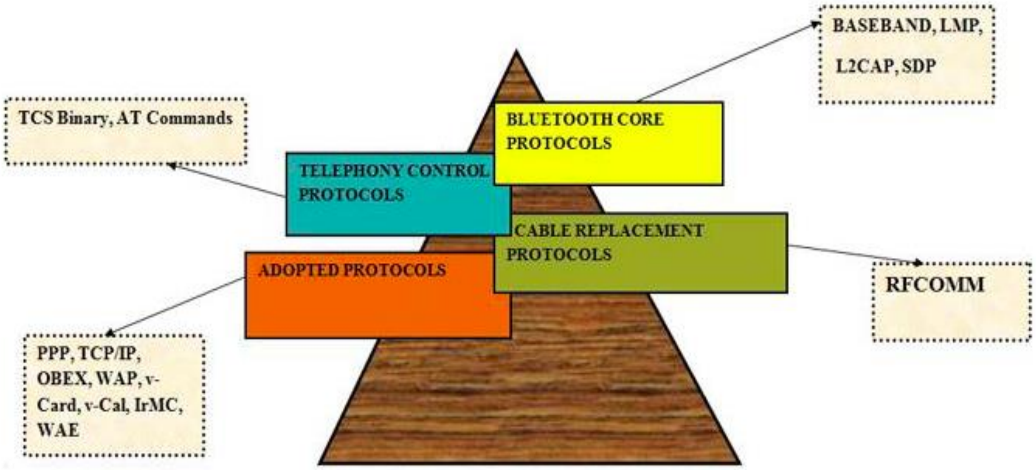


Figure 4.1: Bluetooth® Core Protocols

○ **__Serial Port Profile (SPP):__** This is the profile we will be using for our design, it acts like RS-232 which sends a lot of data between two devices. It builds on top of the serial communication mentioned earlier in that it uses less equipment, transmits data asynchronously. Using this profile (SPP), a device can send and receive information like RX and TX lines connected between at the same rate. Two microcontroller can communicate back and forth wirelessly effortly across a wide range (distance) as if they are connected with wires. Another benefit of this protocol is that it used for integration with any 3rd party application that has SPP profile like an Android phone/tablet and an arduino.

5. Environmental standards

- Since our device is an electronic device. It falls under the standards of electronic industry which includes passive components (resistors, capacitors, inductors); semiconductor components (discretes, integrated circuits); printed circuits boards (single and multilayer boards); and printed wiring assemblies. Our voltmeter will include an integrated circuit (a semiconductor device), thus it will encounter all the problems faced by integrated circuits such as. We divide the standards of the multimeter based on the components in the device as follows:
  - **Semiconductors:** are produced by treating the material with dopants(boron and phosphorus), other materials used are silicon and gallium arsenide. the process involved in manufacturing semiconductors produces toxic chemicals which are harmful to the environment as well as the user. during the manufacturing process such as crystal growth; acid etch; doping and oxidation; diffusion and ion implantation; vapor deposition all release chemicals which are dangerous to one's health. thus it is necessary that we purchase and use semiconductor materials that have gone through a proper procedure in the manufacturing processes , the reason being simple. a material that's made from highly quality standards is safe for both user and the environment. The biggest problem with production process is the **Air emissions** that happens. Chemicals that they use included the following: hydrogen, silane, arsine, phosphine, diborane, hydrogen chloride, hydrogen fluoride, dichlorosilane, phosphorous oxychloride, and boron tribromide, all of which extremely dangerous. Once the production is complete, the leftover include heavy metals, waste organic solvents etc. So the production chain of the semiconductor is very crucial to us. when purchasing these materials they must meet the Electronic Manufacturing process for environmental guidelines. thus the production of semi conductors should be carried out in a closed system.
  - Printed circuit board (PCB); these boards come in three parts, (single sided, double sided, and multilayer boards). board production occurs when you produce patterns of conductive material on a non conductive substrate (coating layer) by subtractive and additive process. the production process releases chemicals as well like sulfuric,hydrochloric, phosphoric, nitric and acetic; chlorine; ammonia; and organic solvent vapors . waste produced after the manufacturing of PCBs are organic solvent and heavy metals.On top of that, the process involves cleaning and surface preparation of the base; electroless copper plating; pattern printing and masking; electroplating and etching; all of which release hazardous chemicals. therefore it is very crucial that PCB go through the same guidelines of semiconductors. which is using chemicals or solvents that will minimize the release of wasteful material.
- Producers/manufactures of PCBs and semiconductors must go through correct and safe processes. they must follow the Environmental assessment (EA) process which are based on the country legislation in our case USA) and the *pollution prevention and Abatement Handbook* as applied to local

conditions. the chemicals used must be justified to the EA which oversees what goes into these materials and acceptable to MIGA.

- ROHS Guide: stands for restriction of hazardous substances. these are guidelines thats makes sure the materials used in electric components are Lead-Free. These guidelines were set up to protect the environment and users from inhaling harmful chemicals. anyone producing, selling, distributing must make sure there products are compliant with the ROHS guidelines. the specification are the following:

ROHS Restrictions	
MATERIALS	**QUANTITY ALLOWED**
Lead (Pb)	<1000 ppm
Mercury (Hg)	<100 ppm
Cadmium (Cd)	<100 ppm
Hexavalent Chromium(Cr VI)	<1000 ppm
Polybrominate Biphenyls (PBB)	<1000 ppm
Polybrominated Diphenyl (DEHP)	<1000 ppm
Benzy butyl phthalate (BBP)	<1000 ppm
Dibutyl phthalate(DBP)	<1000 ppm
Diisobutyl phthalate (DIBP	<1000 ppm

Table 4.2: ROHS Restrictions.

ROHS Categories	
CATEGORIES	**DESCRIPTION**
1	Large household appliance: refrigerators, washers, stoves, AC
2	small household appliance, vacuum cleaners, hair dryers, coffee makers, irons
3	computing & communication equipment: computer, printers, copiers, phones
4	consumer electronics: TVs, DVD players, stereos, video games
5	lighting: lamps, lighting, fixtures, light bulbs
6	power Tools: drills saws, mail guns,
7	Toys and sports Equipment videogames, electric
8	medical devices and equipment
9	Control and monitoring Equipment
10	Automatic dispensers: vending machines, ATM machines.
	National Security use and military equipment
	Large stationary industrial tools
	spare parts for electronic equipment in the the market before July 1, 2006.
	Certain light bulbs and some batteries

Table 4.3: ROHS Categories.

- Waste Characteristics: are the chemicals that are released once semiconductors and PCB are no longer in the factory but are in the hands of consumers. We will talk about the most wasteful chemicals released by parts of the device we are making. this section will be based on the most wasteful “types” produced by the voltmeter we are making.
  - Air emission:
  -
- WEEE: stands for waste From Electronic Equipment, they mandate or demand that electronic products that have been discarded be recovered and recycled. All applicable products (electronic components) must pass the WEEE compliance and carry the “wheelie Bin” sticker.

4.1.2 Software Standards

Most of the software standard will be focusing on the communication between the device and the Android application that will be used for displaying our measurement output. They will be two parts to our standard, the first one will be for the microcontroller and the second part one will be for the Android standard.

The microcontroller that we will be using is the MSP430, this embedded system will have Bluetooth® capabilities in order for it to communicate with the Android application. It will use Bluetooth® 2.1 which will be complete validated, certified and production ready modules. The MSP430 will come pre integrated with Bluetooth® software stack, it will include full hardware and software prototyping, the software kit includes Bluetooth® software stack and serial port profile (SPP). It shall follow the

- FCC/IC regulatory compliance
- FCC part 15 class a complaint
- IC ICES -003 Class A Complaint
- Bluetooth® core specification 2.1+

For the Android application, we want to make sure it communicates very well with the microcontroller. That is why we shall use standards that work and communicate with the hardware (MSP430 stack).

- Bluetooth® core specification.
- API interfacing with sensors on the microcontroller
- Transfer protocols such as serial communication

4.1.3 Impact of Standards

The impact of implementing these standards is that it will be able to make the software on the Android interface and communicate with the microcontroller on the multimeter. With proper communication the microcontroller should seamlessly transmit data with the Android software and display the results. The implementation of these standards across different platforms enables easier communication between the two mediums.

4.2 Realistic Hardware/Application Design Constraints

As the name suggest , constraints are limitation that we have to work with. We as engineers always have to work with what is available and work around obstacles to achieve what we want. In our process of making the voltmeter, we shall be work within the confinements of the electronic equipment that we will be using. The confinements cover all the components from microcontroller to the circuits board; as well how much time we need to o build our voltmeter. A lot of time will be spent on the constraints of microcontrollers because of how crucial it is to make for the success of this project. Also because of the difficulty will are predicting when we make the connection between microcontroller and the Android application. So it's very important that we have a full understanding of the limitations we will be working on.

4.2.1 Microcontroller Constraints

The microcontroller is the piece that connects everything in our system. it will act as the bridge between the integrated circuit board and the Android application. without the microcontroller, nearly half of our project will not be able to work and the goals set will not be reached or accomplished. However this device comes with its own confinement that will must work with. After doing research and looking at all the microcontrollers available, we decided to settle on the ATmega328P microcontroller because of its great support for Bluetooth® connectivity with other electronic components (our Android phone and custom made application that will be used to display the output of voltmeter readings). For example we need to know the clock speeds of the board, sampling frequency of the data that will be transmitted to the Android phone and why it matters for our design., these are some of the examples and constraints we have to work with that we shall cover below.

- **ATmega328P:** Is the microcontroller that we shall be using for our project. It is low powered device that **operates at 5V**, it can receive input voltage in the range of **7-12V with capability to reach 6-20V.** It has 14 for digital I/O pins, 6 provide PWM (pulse modular modulation) output and the other 8 pins/channel are for analog input. Most Arduino microcontrollers have an 8 channel 10 bit analog to digital converter (ADC), which is 8*10= 80 bits transmitted total at any given time.
- **An ADC:** is used to convert an analog voltage that is continuously varying (known as a signal) within a known range into a digital value . The analog value is a representation of a real world measurement which is the voltage measurement of our voltmeter. The digital values are collected through a process known as **sampling**, this is a process by which arduino board reads/samples the signal a certain time per second which is known as the **sampling frequency.** The graph below is an example of a analog signal. As shown below the green line shows the analog signal rise and fall between -5 and 5 volts. It is a continuous line because that represents the measurement of the voltmeter voltage. The blue dots on the graph are points in time where

the ADC is used to convert the current point of the analog signal into a digital value. the number of times we take a reading per second is the frequency of the sampling.

-

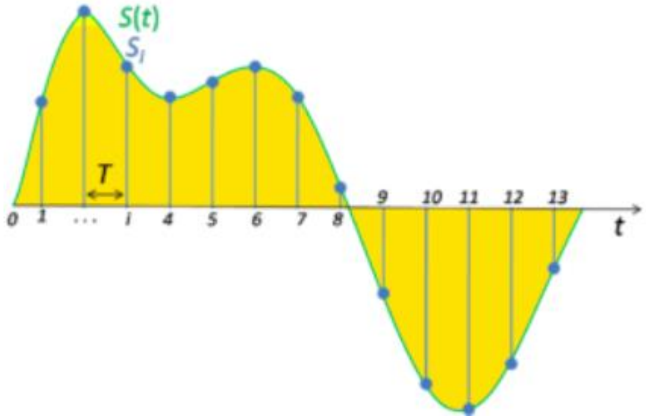


Figure 4.2: Example of sampling frequency (voltage vs time).

- The ADC architecture : is made up of an 8 channel multiplexer. The multiplexer combines the 8 analog pins into the single 10 bits ADC v.Only one ADC operation can be carried at time. If more than one ADC (pin) is being used, the reading/samplings are handled by the **ADC unit**. Ths unit handles which pin use being used , nor two pins can be used at the sample time. if analog pin 0 is being converted, analog pin 1 needs to wait for pin 0 to finish before it can be converted and so forth until all the 8 analog pins have been converted and then it starts all over again with pin 0. The AVR is an 8 bit microcontroller. The ADC has 10 bit resolution. The result of the ADC conversion needs to be stored in two registers. These are ADCH & ADCL (ADC high & low).
- The good thing about this board is that it is user friendly, particularly when mounting it to a breadboard which was a main reason why it appealed to us. especially when working with The device will receive data from the integrated circuit in the form of an analog signal (it has 8 analog input pins), it will use the signal from the board and convert it to a digital signal at rate (clock speed of 16MHz) defined by the microcontroller specifications.

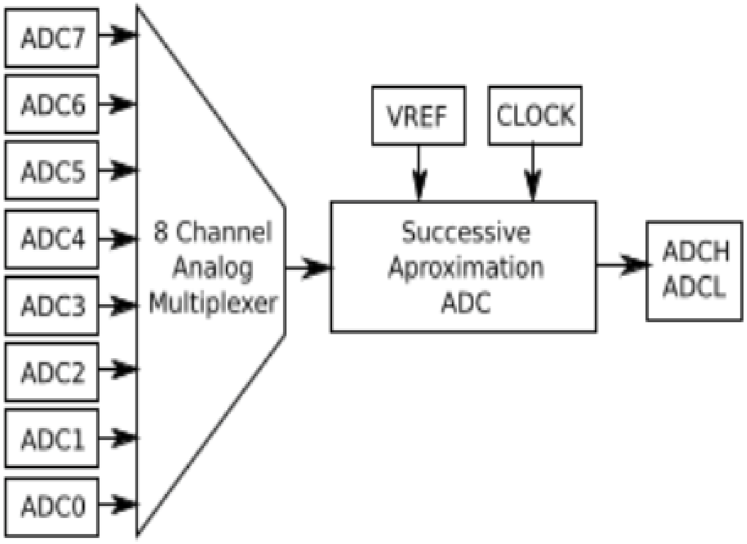


Figure 4.3: Analog-to-Digital Converter

- The input “VIN” of the ADC must be between 0 volts and the ADC reference voltage VREF which is default to VCC and on MCU which is 5 volts, although this can be changed to any external voltage we desire using the AREF pin. Vin is the measurement from voltmeter that is connected to one of the ADC input channels. The ADC conversion formula is **((VIN/VREF)*1023)**. if VIN= 0 volts (assuming VREF= 5V) then the converted value will be 0. if VIN=VREF then the converted value will be 1023 which is the resolution.
- **Operation Modes:** The ADC has two operation modes. The first mode is called the single operation mode. in this mode you have initiate each conversion, once the conversion are done it is stored in the ADCH and ADCL registers. You can begin this process setting the ADSC i.e ADC start conversion but in the ADCSR. What it will do is that it will stay high while the conversion is in progress and will automatically be cleared when the conversion is complete. there is also an option to select which channel to convert the signals. Once it done, you will get a notification conversion completion event can be obtained via the ADC conversion complete interrupt service routine (ISR). The mode is called the Free Running Mode. In this mode we initiate the first conversion and the from there the ADC will automatically start the next conversion. Like the single conversion mode, a new channel conversion must be set in ADMUX before the next conversion begins. be careful not to change the ADMUX just after a conversion starts. This can lead to unspecified behaviour.
- The ADC has clock speed which is between 50 KHz to 200 KHz when 10 bit resolution is desired. This is equivalent to 50,000 to 200,000 per second. Ay faster than this range and the system will being to degrade which is why it comes with a prescaler, this is a component which controls the ADC clock. Its special counting circuit which is used to reduce high frequency electrical signal to lower signal using integer division. the prescaler takes the basic timer clock frequency of the arduino and divides it by some integer value before feeding it to the timer. The prescaler values can be configured to the

desired number. But are fixed to values of powers of 2 (2,4,8,16,32,64,128). For example, suppose the arduino is running at 64 MHz and we want to reduce the clock speed, we will set the prescaler to value of 4 bits, then the clock speed will reduce to 60MHz/4 = 16. The reason why we do this is if we want to to sample analog signals at slower rate in order to calculate the data appropriately. they can act sytheinzier from the circuit board to arduino and then the android application. there is possibility that we might sample signals so fast that they are not properly displayed because they change quickly. However the problem with undersampling is they we might not get the correct digital replica of the signal, so we shall over sample our analog signal to get the correct analog signals. **We shall use Nyquist Sampling to get the correct sampling of the signal.**

- **Resolution:** the ADC has 10 bit resolution. This translate to values between 0 and 1023. If the reference voltage is 5 volts, this means that the smallest detectable change in voltage on the input pin is 5V / 1023 = 0.0049 volts or 4.9 mV.
- Once the Signal is in the microcontroller, it will be converted to a digital signal. This will happen during switching i.e. switching regulators in the circuit that will flip from “ON” to “OFF” voltages that translate to 1 or 0. Essentially what it does is, as the power voltage is leaving the circuit board and about to enter the microcontroller , the voltage at terminal output is changed to digital signal made up of 1’s and 0’s.The reason why it will be converted to a digital signal is because this domain is easier to work with on the microcontroller and also easy for transmission of data packets to the android phone (the phone does not have an analog or FM receiver/ tuner. Current technology use Bluetooth® technology which allows transmission of data wireless. As mentioned earlier both components need to have Bluetooth® capabilities in order for the transmission to occur, this means that both devices must have chip called DSP which stands for digital signal processor. This is a special chip handles signal processing.
- Once the signal is in the MCU’s data storage area. It has to be transmitted to android phone. The data is in located in the flash memory of the arduino in the form of data packets (10 bits streams of data). basically they are 3 types of memory pools in the arduino boards.
    - <u>Flash Memory</u> - also known as the program space, this is where the arduino sketch is stored.
    - <u>SRAM (static memory random access memory)</u> - is where the sketch creates and manipulates variables when it runs
    - <u>EEPROM</u> - is where the memory space that programmers use to to store long term information.
- **Flash and EEPROM memories** are nonvolatile. This means that the information stays when the power is OFF . Meanwhile SRAM is volatile and will be lost when the power is recycled. The is not enough SRAM in the arduino, so we as programmers have to make sure we do not use all of it. This can happen when use long storage data variables when we storing small values. It is important to free memory after we are done doing a

specific task so that there is room for the next instruction. When the arduino runs out of memory the program may fail in an unexpected ways. It might appear to run successfully but it will run strangely. To see if this is happening, we check out the sketch and try commenting out or shortening strings and other big data structures. If the program run successfully after commenting out, this means we ran out of SRAM. the SRAM is too small to handle big computations. In this case, we move or shift the data to android/phone which has bigger memory pool and perform the calculations there in real time. This will reduce the load on the Arduino and we won't run out of the SRAM. Another option is to transfer the data and store it on the arduino flash (program) memory using the **PROGMEM** keyword.

- Like mentioned previously the data is either in the SRAM or flash memory. At this point we need to transfer the data (digital signal) to the android phone and then it has be displayed in the android application we are building. We have to find a mechanism to transfer that data and display it in human readable form. In the arduino memory the signal is just streams of 0’s and 1’s (binary data) in packets of 4,8,16…..128. These streams of binary data now need to be transferred through a Bluetooth® channel. we accomplish as follows;
  - Bluetooth® serial term module to connect to the arduino (JY-MCU).
  - create a sketch (just another name for the program we writing) and upload it to the arduino and run it there.
  - the first thing we do is set up the the connectivity between the arduino board. We going to to pair the arduino with the android phone. we will have a check-connectivity, a way of verifying that a connection as has been established between the two platforms. one way of doing this is 1) have the arduino flash the LED lights on the board, 2) display “Arduino name” on the phone screen to show that the pairing was successful.If no connection has been established, we shall reset the arduino mechanically on the board and start the whole process again. the benefits of doing such is that whatever is in the volatile memory will go and have a fresh startup. This step is very important, if no connection then we cannot even get the reading from the data.
  - <u>Data Transmission Via Bluetooth®</u> - Data transfer depends on the protocols, these protocols are designed based on how much power the devices produce and use. The devices arduino and phone roughly have power consumption of (5V*46.5 mA = 232.5mW and 1 KWh respectively). the Bluetooth® module has two mode of operation. **Another important factor we have to consider is that the android phone might have different baud rate to that of the phone**
    i. **<u>Command Mode:</u>** in here we send At commands to it.
    ii. **<u>Data mode:</u>** This is the default mode for the Arduino.In this mode we can transmit and receives data to another Bluetooth® module. Based on the data sheet, at 16 MHz it transmit 1 Mbps.
      - Baud rate: 9600bps of data: 8 bits, stop Bits: 1 bit, parity.
      -

  - Display the Data - depending on which configuration we used (whether we calculated the distance)

4.2.2 Economic and Time Constraints

A lot of time will be spent on making sure the Android application communicates with the microcontroller on the voltmeter. Finding a way to make the protocols to make the two machines to work together will be the biggest obstacle. The reason for this is because these two machines are used for different purpose but they can work together thanks to protocols that enables two devices to work together. so a lot of time will be spent on finding the correct microcontroller that will communicate with the Android device. A lot of time will be spent on making sure the sensors work well and are able to transmit the data to the Android device and in turn the device is able to communicate back to the microcontroller on the voltmeter.

Another area where we will spend a lot time is on the testing of the circuitry to make sure we are able to measure the voltage accurately and precisely; the microcontroller and its properties like the sampling rate and frequency it will transmit the data, If we are not able to make measurement because of the design of the circuitry or unable to transmit the data to the Android application, then we will have to redesign the circuitry and then retest it again until we get the correct measurements.

Thus it is possible we might spend more money on the circuitry and microcontroller if our initial assumptions are wrong

4.2.3 Environmental, Social, and Political Constraints

Since this is a portable device, it will be used outdoors a lot. we want the device to be environmental friendly and usable. It should be used by for its intended purpose and no nothing more. Since our device uses CAT 1 standards, it will be limited to what is specified in that category which handles electrical environment with high power available and higher energy transients.

The device shall be made with reusable material and reduce emissions of greenhouse substances. It shall be designed in accordance to the certification agencies which evaluate that the system(s) implemented by us is safe to both the user and the environment environmental factors such as storage temperature, humidity, air density and electromagnetic radiation can affect uncertainty, thus it is imperative that the device must receive clean power.

On top of that, we expect users to use the device in the right environment and for the right purposes or intentions. This means we do not want anybody to use this device to harm others or modify it. We intend this device to be used by everybody but safely.

4.2.4 Ethical, Health, and Safety Constraints

Besides designing a working multimeter, safety is also very important for our customers. A device that is safe to use in the hands of our customers is very important to us. We do not want any hazards nor electoral shocks happening. This is why we have taken greater measure to make sure that our device is very safe, stable and reliable. To do this we will be following industry standards in accordance to our machine. We shall implement the “category 3” for our measurement category because it is not connecting to MAINS.

- One of the hazards likely to occur is electrical shocks from the device. Despite it being low power consumption device, there is a possibility of it electrocuting the user. That is why we have to make sure the device has an insular cover to prevent the user getting an electrocuted. Voltage spikes are likely to occur in situation where a user is measuring electrical systems that use high energy transient from transmissions or high energy circuits. One standard that we will be using is ***“IEC 1010 standard”.*** We will use the guidelines to test against high energy circuitry.
- **Transient protection** is another important area against against steady state and transient over voltage. High energy circuit with transient ride can deliver high current voltage which can cause an explosion.
- **Overvoltage installation** is another important aspect that must be taken care of. for our device we shall implement “CAT III”. which insure that the low energy voltage is derived from high energy high energy circuits.
- **Overload protection** against overcurrent might be prevented when adding high impedance of volts/ohms. It is achieved by protecting circuit that clamps high voltage to acceptable level.

4.2.5 Manufacturability and Sustainability Constraints

As mentioned in section 2.4, this multimeter will be low powered device, that will be reusable, use eco friendly parts. The device should last long during operation (around 200 hours). It should be easy to assemble to make mass produce. The device should use the best parts at low prices since it will be used different purposes. It should be kept under correct weather conditions such as temperature, humidity, Air. This device will be reliable, durable and affordable.

5 Hardware and Software Design Details

In this section, we will discuss in detail how our product design began, the issues and concerns that came about, and how we came to our final design. We discussed in section 3 how we chose the parts to be used in our product, and here we will discuss how they will be implemented together to meet the requirements specifications defined in section 2.

5.1 Initial Design Architecture

As discussed in section 3, there are vast amounts of information to base our initial design on. From the basic pocket multimeters to advanced Fluke meters, there are many paths to choose in designing our product. From the beginning of our project, we needed to make a decision on if we had the knowledge and experience to achieve our goals on both an electrical engineering and computer engineering basis. To determine this, we needed an initial design to give us an idea on where we stood. After many hours of researching various products and designs, we decided to take a stab on the breadboard and attempt to take a basic voltage measurement using the Arduino Uno.

5.2 Initial Voltmeter Design

Our first design was based off of the general properties of the analog-to-digital converter used on the Arduino Uno. This component is 10-bit, which gives a resolution of 2^{10}, or 1024, discrete values. For our requirement specification to be met, we need to be able to accurately measure voltage to the hundredth of a volt. This component operates on the Arduino's 5V reference voltage which gives the voltage resolution to be 5V/1024 = 4.88mV which is sufficient for our product requirement.

Once we had established that the microcontroller could give us the resolution we required, we had to address what voltage we needed to measure. We decided to strip the ends of a 12V DC wall adapter and use this for our initial measurement test. The only issue this decision presented was the input limitations of the ADC. An ADC, like any integrated circuit, operates within a range of input voltage. The ADC on the Arduino Uno is contained within the microcontroller itself, the ATmega328P, which is set to a default reference voltage of 5V. At this point, being that this test is a only a proof of concept, we were not concerned with being able to measure a wide range of voltage. Through the registers inside of the ATmega328P, we can alter the reference voltage that the ADC operates on. This will be altered in our final design to optimize the desired measurement range and maximize the resolution of our product. Our power supply output, being 12V, violates the input range for this component to operate correctly and must therefore be dealt with using a voltage divider. We wanted to be able to safely measure a range of around 30V for our initial design, so using the voltage divider equation, we chose resistor values of 4.7kΩ for R1 and 1kΩ for R2. These resistance values give us an ideal input range of 0-28.5V to the divider and a 0-5V input range to the ADC.

Before we could begin implementing the basic circuit, we needed to take measurements of the components we would be using and testing. Using a store bought Craftsman digital multimeter, we measured the voltage across our makeshift power supply to be 12.03-12.04V. We then measured the actual resistance of R1 and R2 to be 4739Ω and 988Ω respectively. These values are critical for our design to operate correctly, as the conversion from digital values out of the ADC back to the original input voltage is done through software. We then connected the circuit on the breadboard and began writing the code using the Arduino program.

We began our code by declaring variables such as the value read on the output of the ADC, our resistor values, and other temporary values to be used in the calculations. The program reads the output of the ADC and multiplies this value by our resolution. This gives the actual voltage that is being read on the input of the ADC. This value is then placed into the voltage divider equation using the actual measured resistance values stated above. We then utilized the Arduino's serial functionality to read, calculate, and display the value on our computer's serial monitor. The code then hits a delay and goes on to repeat. We added a few other measurements such as the average voltage, time elapsed, number of samples, and maximum/minimum values.

5.2.1 Breadboard Testing

Once our program was complete, we probed the output of our power supply and found that our barebones design was correctly reading voltage within 0.01% of the store-bought meter. This was well within our accuracy requirement of 0.1%. We then added a second voltage divider in series to vary our "power supply" voltage to be measured. For the range of 0-12V, the design worked flawlessly and within our accuracy and resolution requirement specifications. From this initial test, we confirmed that we have the tools necessary to carry on with our project and construct a functional voltmeter.

5.2.2 Issues and Corrective Measures

The first noted issue with our design is our range of measurement. As stated previously, our range for measuring was 0 to 28.5V with this set up. To meet our requirements specification of a range from 0 to 1000V, we needed to redesign our voltage divider to allow such a range. On the same note, our second major design flaw arose; the input impedance of our meter. Our highest concern besides user and product safety is to not disturb or affect the circuitry being measured in any way. We want to simply read a voltage without causing any changes to the circuit being measured. We also want to ensure that our meter is accurate for voltage measurements on any electrical circuitry, whether it be a power supply or a component in a system. The input impedance of our voltmeter is the key factor to addressing this issue.

In order for the meter to correctly measure a voltage, we want the smallest amount of current to flow through our terminals. In order to do so, we have to make the input impedance of our meter as large as possible. Most modern voltmeters have an input impedance of 1MΩ (Category I and II) and upwards of 10MΩ for higher rated meters. The ADC itself has a very high impedance, around 100MΩ on the ATmega328P, but this does not ensure the safety of the device. To ensure the safety of both the user and our meter, we have to create a very high impedance between the probes. We can accomplish this by adding several large resistors in series to the input of the ADC.

To address the range of our input voltage, we need to discuss the range switching of the voltmeter itself. Because the voltage drop being fed into the ADC depends itself on the input voltage, we cannot simply use the same two resistors to get an accurate measurement for both small and large voltages. To work around this issue, we have to program the microcontroller to select the correct path for the range of voltage in which we intend to measure. Modern meters use complex designs between the physical dial and the PCB to select the correct path, along with multiplexing inside the main IC. The Arduino Uno contains six analog input pins which are directly fed into the ATmega328P which we can use to our advantage. By wiring the circuitry into the different input pins, we can create a network of resistors for our different voltage ranges. In our case, the ATmega328P has internal multiplexing technology that selects the correct channel to be used.

The next issue to be addressed is the optimization of the ADC. Using the default reference voltage of 5V means that we can safely measure from 0 to 5V with the analog input pins. As mentioned before, this gives a resolution of 4.89mV, meaning our measured voltage is no more or less than this value. The problem now lies the accuracy requirement of 10mV for our meter which is discussed further in the next section. As stated previously, we want to be able to safely measure a wide range of voltages, ideally up to 1000V. Using a network of resistors for the different voltage dividers for different ranges, we can control the inputs to the ADC to be on a much smaller range. By doing so, we can adjust the reference voltage of the ADC to match this range and give us a much higher resolution. This is explained in further detail in section 3.

The final and most important issues to address are the safety hazards of our current design. Input protection to the device is absolutely critical for both the protection of our product, but more importantly the safety of the user. Because this test is only a proof of concept and we were only measuring 12V, protective circuitry other than the voltage divider was not necessary. In our final design however, the safety features implemented will be the most important and crucial part of our project. Our safety design and components are outlined in section 3.4.6.

5.3 Updated Hardware Design

Because the initial test was prior to our research and parts selection, our second design had to be vastly changed. Once our parts had been ordered, we drew the schematic for our updated design which can be seen in Figure 5.1. As shown by the schematic, we selected our nominal input impedance (R_1) to be a single 10MΩ resistor which we knew would provide us with a high input impedance. For the input protective circuitry, we used a total of four components. The first component we knew our meter would need was a single large varistor. This component would act as an open circuit during normal operation, but would clamp down when the voltage being measured was above the varistor's clamping voltage, shunting the current through back through the negative test lead. To handle this large current, we chose to use a PTC thermistor and a fast-blowing ceramic fuse. The resistance of the thermistor would rapidly increase due to the heat produced by large current surging through the component. The fuse at the front of the circuit would blow once the current limit was reached to open the circuit and protect the meter and the user. Additionally, we included a high-power, wirewound resistor that would absorb the large amounts of energy created by the overload current.

In our proof-of-concept design, we used the Arduino's 5V operating voltage as the reference voltage on our ADC, giving us a resolution of 4.89mV. This was fine before because our voltage divider values were so low that a 10mV deviation on the input provided a voltage drop within our resolution. Now that we have a much larger input impedance and thus a larger range, we need to compensate by maximizing the resolution. The ATmega328P, as mentioned previously, has the option to set the ADC reference voltage (AREF) to any desired voltage. In section 3 we calculated our R_2 values based on a reference voltage of 1V. In order to maintain our range with these R_2 values, we had to set the AREF pin to 1V. We therefore connected voltage divider from VCC to AREF with R_1 and R_2 to 40kΩ and 10kΩ respectively to give a 1V input to AREF.

In this design, we had four resistors in our voltage divider network to have a range that could potentially measure up to 10kV. Each of the resistors were connected in parallel with a 4-pin DIP switch between the resistors and ground as can be seen in Figure 5.1 below.

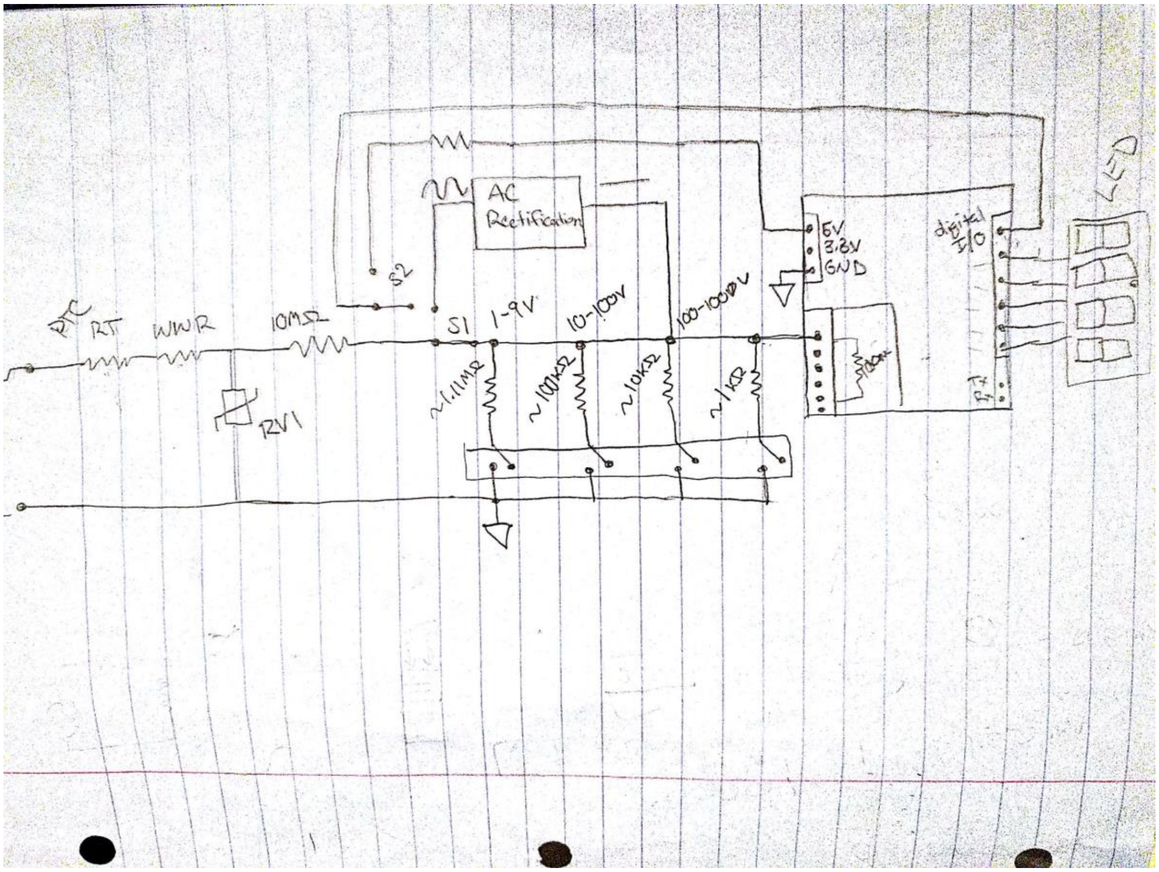


Figure 5.1: Updated Design Schematic Layout

The node in which each of the resistors were connected fed into a single analog input. For measuring in the millivolts range, the user would not select a range switch as the voltage would be directly fed into the single analog input. Additionally, we had a separate 2-pin DIP switch in which both of the switches were tied together for selecting AC or DC measurement. One of the switches re-directed the raw input voltage to an AC rectifier circuit which would then send a DC voltage to the resistor network to divide the voltage to a scalable input to the ADC. The second switch connected 5V to a digital input on the microcontroller which would set a flag in the code and jump to a separate loop that calculates the true RMS voltage. With these switches connected together, we could ensure the user would get an accurate measurement for both AC and DC modes.

We also decided to include a digital LED display would be connected to four of the digital output pins of the Arduino. Although not displayed in the schematic, this design also included the Bluetooth® module to be connected to the RX and TX pins of the Arduino for wireless transmission of data.

5.3.1 Discovered Issues

After further research and scrutinization of our updated design, we found several flaws and potential issues that would hinder the quality of our product. The first issue we discovered came upon simulating the measurement using a computer simulation software. The design appeared to be working correctly for each of the range selectors when only one range was selected. The problem with this design is when none of the switches had been selected. In this case, the voltage would be directly fed into the ADC. We saw that this design could potentially feed large voltages into the IC which would damage the product. This led us to the discovery of our second major flaw. Because we were using a single analog input, the microcontroller had no way to realize which range the user had selected. This means that the voltage computation would use the same parameters for all ranges which would result in incorrect measurements. To fix this issue, we realized we would need four separate switches that would need to be engaged simultaneously with the specific range selector. This means that we would need to use four additional I/O ports on the microcontroller.

The final range selection issue was the possibility for the user to select more than one range switch at a time. In this design, if two or more switches were selected, the resistors would be connected in parallel which would result in inaccurate measurements and conversions. To remedy this, we could potentially use a switch that only allows one button to be selected at a time. Unfortunately, this would prevent us from implementing the option mentioned above of connecting a second 4-pin switch for the MCU to know which range we are operating in.

The second major issue we noticed was the correlation between our potential measurable range (up to 10kV) and the ability of the safety components to allow this range. After further researching several models of varistors, we could not find a component with a large enough clamping voltage to allow such a range of measurement without activating. To fix this issue, we would have had to use around 27 smaller varistors (365V) in series with one another! This would not be realistic as the large number of components would take up huge amounts of space on our PCB and increase the cost of our meter dramatically. We then realized we had to reconsider the realistic range in which our meter could measure without activating the varistors.

On a related note, we also realized a flaw in our input resistance. Although our chosen 10MΩ resistor would provide enough input resistance, we had to address that this was only the “nominal” input resistance, meaning that the safety components would inherently add to the total resistance as seen from the input. This would then affect the calculated voltage while also changing the actual measurement ranges for each resistor in the divider network.

The next issue arose with our AC rectification design. In this design, our AC voltage would be fully rectified and then converted to a DC voltage using a capacitor across the output. As the rectification would occur before the voltage division, we realized that the capacitor used at the output would need to be specific to the range of voltage it would be rectifying. This means that in addition to our range selector switches, we would need to implement a second network of capacitors for each range. With this new roadblock, along with the necessary addition of the previously mentioned 4-pin DIP switch for the MCU to be able to calculate measurements correctly, we quickly realized how many additional switches we would need to implement this design. While brainstorming options to fix this issue, we discovered another major design flaw with the AC rectification. One of the most important features of our product is to allow the user to view AC waveforms on the smartphone application after being wirelessly transmitted from the microcontroller. For the application to be able to plot this data, it has to know the frequency of the signal. In this design, we fully rectify the signal before being input to the ADC so the microcontroller had no way to determine the frequency of the measured signal. As a possible corrective measure for this flaw, we found that the Arduino IDE has a function that can measure input frequencies using one of the PWM pins on the microcontroller. If we were to utilize this feature, we would have to tap off of the input to the ADC which could possibly alter the electrical characteristics of the signal being measured due to current leakage and noise on the I/O pins.

The final largest issue to this design was the use of the Arduino platform. We had completed this design while still under the assumption that we would be allowed to use a premade development board from Arduino. After further investigation, we discovered that we are not permitted to use such products as we have to implement all of the hardware ourselves. This meant that we would now have to design and implement the input power supply and voltage regulation, along with the crystal oscillator and programming circuitry.

5.4 Final Hardware Design

After taking into consideration the multiple design flaws of our updated design, we came to our final design which would correct all of the aforementioned issues. To begin this section, we will discuss the measures taken to address the measurable range, the range selection process, and how these measures contributed to the re-designed safety features. We will then discuss our solutions for AC rectification issues and end the section with additions that needed to be made for the purpose of powering the MCU without an Arduino development board.

The first issue we had to correct was the measureable range of our meter. We realized that because our meter is only intended to be used for low voltage applications (<1000V), we did not need a range above this value. We decided to remove the 1kΩ divider which left us with only three resistors in our network. This action now meant that the largest measurable range was 100-1000V. To address the issue of a potentially large voltage being directly fed into the ADC, we had to reconsider how we selected our range. Our first idea was to implement a 5-position switch between the nominal 10MΩ resistor and each of the range switches. In the top position, the circuit would be open and no voltage would be fed into the ADC. With this design, we would now have to use a separate analog input for each range. While this seemed to be a disadvantage at first, we quickly realized that this solved a few of the other discovered issues. By using a separate analog input for each range, we could now write code that would detect which input was being used. Therefore, we can eliminate the need for a second row of switches to tell the MCU which range we had selected as it can now be sensed automatically. Additionally, with this design, we can write a loop that checks to see if more than one of the analog inputs is detecting a voltage. If more than one input has a voltage, as would be the case when more than one range selection switch is enabled, we can send the user a visual or audible warning using either an LED or a piezo speaker.

After we had decided this was the route we would take for the range selection process, we began to search for the 5-position switch that could realize our design. What we quickly realized was that these types of switches were very large and not intended to be used for PCB applications. After once again reconsidering our switching options, we found that the 4-pin DIP switch from our previous design could still be used. We found that by placing the switches in between the 10MΩ resistor and each of the dividing resistors, we could now have the entire circuit open without the need of a fifth position path. This is also due to the fact that we eliminated the 10kV range and resistor. The final switching circuitry is shown in section 6.

With the range selection sorted out, we can move onto the issues related to the protective circuitry components. Now that we have decided our measurable voltage range would be a maximum of 1000V, we can address the necessary components to protect from high voltage and current overloads. The first component we will address are the varistors. We found varistors rated with a clamping voltage of 340V. By connecting three of these components in series, we can create a total clamping voltage of 340*3 = 1020V. This will work perfect for our design as is it will conform to the Category III meter rating, along with minimizing the cost of the meter. Additionally, this will save much more space in comparison to the previous option of nearly 27 varistors. The thermistor implemented in the second design will remain for this design as it is necessary for protection against high amounts of incoming current. Additionally, the wire wound resistor will remain in this design in order to dissipate the large amounts of energy that can be produced if connected to high voltage and high power supplies. The only remaining change in safety components between our previous design and this design is the elimination of the HRC fuse. With the thermistor and high power resistor in place, there is no need for the fuse as the PTC will take care of essentially opening the circuit in the case of a high voltage transient.

Now that we have corrected all of the potential issues with the range selection and safety components, we can discuss the issues associated with AC measurement. After considering the several flaws in our second design, we came up with a few possible options to take care of these issues. We knew that we had to somehow measure the frequency of the incoming signal in order to be plotted in the smartphone application. Like we mentioned previously, we could potentially tap off of the input before the rectification and voltage division, but this would mean that the peak of the AC signal must be at or below 5V, which is the maximum input to the digital pin that would count the pulses. This means that we would have to first divide the voltage before feeding the input into both the ADC and the digital pin for frequency measurement. Unfortunately, for the digital pin to recognize the upper portion of the waveform, it must be at least 3.3V which is above the maximum output of the voltage dividers (1V). After exhausting all options for this dilemma, we concluded that we would not be able to both rectify the signal and measure frequency at the same time. As measuring the frequency of our signal is detrimental to the application of our product, we decided to knick the idea of fully rectifying the signal and converting it into a DC signal.

After making this decision, we came up with a few other possible designs. The first option we considered was to not rectify the AC signal in any manner. Instead, the waveform would first be divided using the same resistor network for DC measurement before entering the ADC. Unfortunately, the ADC can only measure from zero to its reference voltage, which is 1V. This means that the negative cycle of the signal would not be read by the ADC. We would therefore have to add a DC offset to the signal to place the entire signal within the ADC's measurable input range. This would mean that the peak-to-peak voltage of the divided signal must be a maximum of 1V, which in turn means that we would have to apply a 0.5V offset to fully place the signal between zero and 1V considering the maximum of each range. In this option, the range of measurement for each range would be cut in half. For example, if we are measuring in the millivolts range, the amplitude of the signal would be limited to only 500mV (0.5V) instead of 1000mV (1V) because of the added offset. This is the same scenario for each of the ranges. The 1-10V DC range would now become 0.5-5V. The 10-100V DC range would now become 5-50V and so on. This now cuts the overall limit of AC measurement in half, which is still a flaw in the design, but it could still work for our project. We spent a large amount of time researching how to implement the DC offset, and after exhausting all designs, we decided this was not the way to go. The first issue came about when attempting to add the offset using a voltage divider where the input is +5VDC and the output is 0.5V. In order to not draw any current away from the signal, the resistors used in this divider have to be very large. The "R_2" resistor is connected to ground on the other end, and therefore changes the resistance of the first "R_2" in the input's voltage divider because they are now effectively connected in parallel. Even using upwards of 100MΩ for the offset resistors affects the input divider. To fix this problem, we considered trying to separate the measurement ground from the logic ground. Unfortunately, we found that the ADC must use the measurement ground as the logic ground for the ADC to operate correctly. Knowing this, we then knew we would have to re-calculate the ranges of input that we could measure.

Because we are feeding four different analog inputs, we would need four "offset dividers". Additionally, before the offset is added to the signal, we have to remove any previous DC offset from the incoming signal using a capacitor so that our measurements will be correct. Unfortunately, if we were to connect a capacitor before being fed into the ADC, then DC measurement mode would be completely eliminated! This means that we would have to connect another set of two-position switches that would have to be selected along with selecting the range, one position feeding directly into the ADC, and the other being passed through the capacitor and offset circuitry before being fed back to the ADC. We thought of possibly using separate nets for the AC measurement, but because our microcontroller has only 6 analog inputs, four of which are used for DC measurement, we would only have two selectable ranges for AC measurement.

This design option had quickly snowballed into a large mess of switches and paths that the user would need to select before even taking a single measurement. Although we could spend large amounts of time scrutinizing our design and optimizing it for accuracy and usability, we had to accept that the physical implementation needed for this would require far more time than we had. It would have been a huge advantage for users to be able to see the entirety of the waveform exactly as it is seen from the ends of each probe, but we had to acknowledge the lack of time we had to design all of the physical characteristics for this to work. After reaching this decision, we had to move onto our second option which is to simply divide the voltage and feed the AC signal directly into the ADC. Although the ADC cannot convert a negative input, it produces an output of zero. Because the incoming signal voltage has been divided already, the maximum, or technically minimum, negative voltage drop would be -1V. This voltage is small enough that it will not harm the ADC nor affect its operation in any way. Then by using strategic programming techniques, we can take control the ADC to take a very short sample at its maximum sampling frequency to collect a quick snippet of the signal; ideally 3-5 cycles. Each value in this sample would be stored in temporary variables which we can access and manipulate. Because the ADC will output zero during the negative cycle of the signal, several strings of zeros will be created and stored. We can then create a loop that narrows down all of the positive values between these strings and essentially recreate the signal. From this data, we can find the peak value of the signal and then compute the true RMS voltage. Additionally, by counting how many strings of zeros are present when the sample was taken at a known sampling frequency and predetermined length of time, we can calculate the frequency of the signal. While this method will surely prove to be a very challenging, we knew that it was possible and that we had to make it work as it was the last remaining option. The only hardware left needed to implement this method is a single pushbutton switch that must be pressed to tell the microcontroller which mode we would like to operate in.

Once we knew that we had to implement all of the circuitry that is used to power and control the microcontroller, we quickly ordered the necessary parts and took them into consideration for our final design. The components necessary are the LM7805 5V regulator, two 10µF decoupling capacitors, a 16MHz crystal oscillator, and two additional 22pF capacitors for the crystal. We ordered these components, along with a P-channel MOSFET that would be used to protect the battery from accidental polarity inversion. Also necessary for correct operation, we implemented a pushbutton switch that is connected to the reset pin of the microcontroller. This will allow us to reset the IC when loading new programs.

The next change that we made was replacing the LED display with an LCD. As we have enough I/O pins to spare, we knew that this type of display would be better for our design. For example, we can now display multiple values of measurement such as maximum, minimum, and average voltage when in DC measurement mode. When in AC mode, we can now display to the user the frequency, peak voltage, and RMS voltage.

Last but not least, we included four LEDs. A green LED will be connected to the power supply to indicate when the device has been powered on. A red LED will be illuminated when more than one range switch has been selected as this would result in miscalculation. The final two LEDs are blue and will be to indicate which mode of measurement the user is operating in, either AC or DC. Each of the four LEDs will need a 220Ω resistor to limit the current flowing through them.

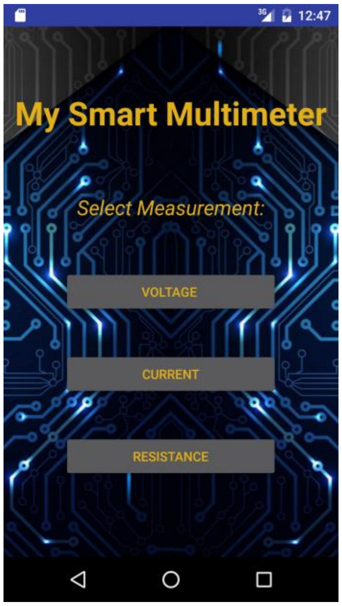
5.5 Application and Final Microcontroller Design

5.5.1 Application

In designing an Android application, guidelines and methods for design, development, and distribution are supplied by the official Android development resources available online. The application will take advantage of the established standard that Android has set in terms of operating system updates. The application is designed to work on 85+% of phones operating on the Android operating system to maximize utility and compatibility. The application should be able to run on multiple devices of varying screen sizes and screen resolutions that encompass the vast majority of Android handsets. Android has performed many changes to its operating system since its inception. One of the more drastic changes is the shift in the graphics of the entire operating system towards a “material design” theme, completely overhauling the home screens, menus, icons, etc. Material design as summarized on the Android developers website is “a new design metaphor inspired by paper and ink that provides a reassuring sense of tactility”. The application will be developed to attempt and recreate this “material design” visual style in order to be inclusive of the Android operating system’s visuals.

The development of an Android application is a fairly rigorous process due to the nature of applications, the scope of operation that applications could potentially cover, and the process at which applications are created themselves. The development tool Android Studio combines 2 different types files, .java and .xml in order to create the visual and logical appearance and behavior of the different activities within an Android application. Activities are basically commands or changes the user indicates the application make based on the input provided. Some activities are default within the Android operating software such as home, back, and recent apps. These activities will continue to operate as normal for this application, however all other activities will be required to be created within the application.

Activities within Android will change based on altering the position of the screen. This poses a design concern because a change of activity restarts the creation process in that activity within the application, resetting all previously calculated and presented data. The application will be able to maintain functionality in both landscape and portrait without resetting data so the user can monitor voltage trends continuously. The user will have the option of resetting the data presented to analyze new data, otherwise the previous data will be stored and visually supplied by graph or numerical value.



During the original design phases of the Android application (See right image) the intent was to display varying categories of information including voltage, current, and resistance using both a graphical and numerical representation which would be able to be changed based on the user's preferences. This original design would give the user the option at the home screen to choose the desired measurement required.

As the design process furthered, the project had been modified to allow for only voltage to be a measurement possibility, this means that current and resistance would be removed from original application design shown above and replaced with a similar menu, however this menu would only allow for AC (Alternating Current) or DC (Direct Current) measurements. Because AC can be represented clearly on a graph depending on the circuit measured, the application will plot the data points measured by the voltmeter and display them in a way similar to an oscilloscope as pictured in the image to the left. (Please note that the image shown is representing a common sin wave as a form of example of the final applications behaviors). The application will also allow the users options of data representation, either the graphical representation as the example shows or a numerical value. Both of these representations will be able to be output or saved so the user can keep the measured values as record for further analysis or reporting. This action will we be done from an external source, i.e the physical voltmeter, and the intent will be transmitted wirelessly in order to trigger the data recording event.

As design of the overall voltmeter has advanced, the application has undergone even more slight changes in order to ensure optimum efficiency of both the hardware and software resources involved. As previously stated in the design section of this document, the voltmeter will physically allow changing of the measurement values of the given circuit. The voltmeter will also allow the user to physically switch between AC and DC voltage measurement. The application will be able to recognize these hardware changes by setting flagged values on the Arduino microcontroller and sending the values to the connected mobile phone. The communication between the application and the voltmeter will be client-server relationship, where the client is the Arduino microcontroller and is managing the measurements done on the circuit and calculating said data into digital values that can be sent wirelessly. The server of this relationship is the mobile device where the application is loaded onto it. The application will read the incoming data and do basic calculations in order to present the user with the appropriate values based on the settings selected by the user, i.e Current type and numerical place value. The AC measurement will present the measured data using the android GraphView library.

The android GraphView library is a third party library specifically created to allow datasets to be represented graphically without manipulating existing android libraries. The Graphview library will be imported to the project and the data points will have to be set dynamically by the values received from the voltmeter. The application will also be able to read the number value whether it is in millivolts, volts, etc. and change the graph headers accordingly.

Below is an image of the final software design tree representing the logical paths taken throughout the use of the application:

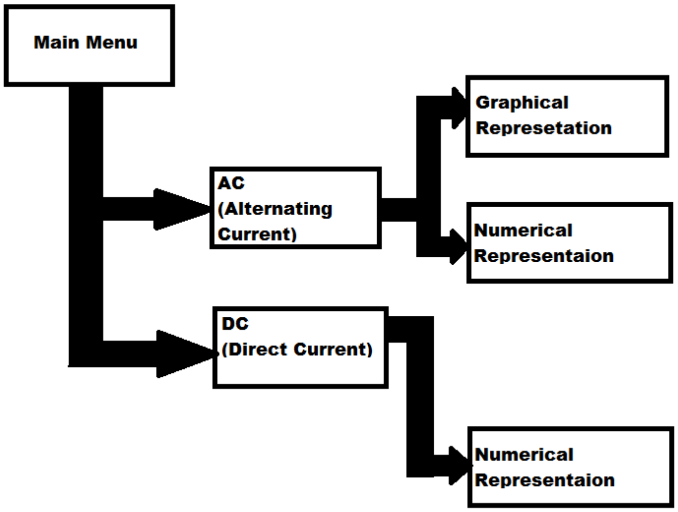
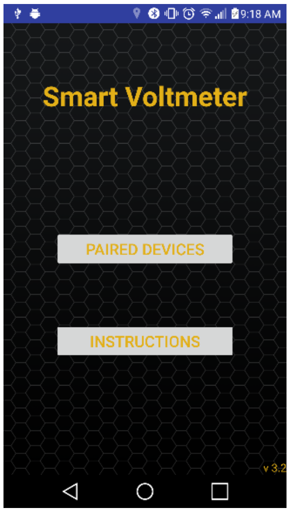
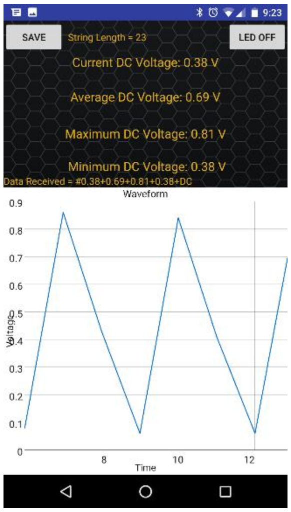


Figure 5.2: Final Software Design Tree

Updated Application Issues and Design

Many iterations of the Android application were completed before deciding on the final design. The main issue in creating a fully functional Bluetooth enabled Android application is due to the life cycle of Android activity's life cycle. When an application's screen is changed, the information and commands on the previous screen are cleared to create available resources for the new activity. This means that connected Bluetooth devices would have the information discarded unless properly handled. This greatly impacted the design of the application and allowed a more streamlined and user friendly design. Major components of the original application were kept in the final design; however, the views were changed to allow the user to view both the numerical and graphical interfaces on the same screen. This keeps the Bluetooth data intact for the duration of the testing and data representation. The image below shows the updated design with the new data representation screen. The button to save the data to a text file is still implemented and functional as in the initial design.





String Length = 23

Current DC Voltage: 0.38 V

Average DC Voltage: 0.69 V

Maximum DC Voltage: 0.81 V

Minimum DC Voltage: 0.38 V

Data Received = #0.38+0.69+0.81+0.38+DC

5.5.2 Microcontroller

The first part of this section deals with the flow and structure of the program. The second part of this section deals with internal flow of the code in the MCU, we shall pay more attention to the MCU control structure since it very vital because we need display what we are measurement.

Section 1: this is how the program will go from once the voltage has been measured. **in the diagram below, the measurement of voltage is represented as sinusoidal wave for simplicity and visual representation**. Now with that said let’s talk about how this will work. The voltmeter will measure the voltage from a source and then at the same time it will convert the voltage it to a digital signal in conjunction with the Arduino. The Arduino will perform the necessary computation that we need and want. When it has done this, it will display them on the LCD then transfer the new calculated information to an android smartphone via Bluetooth

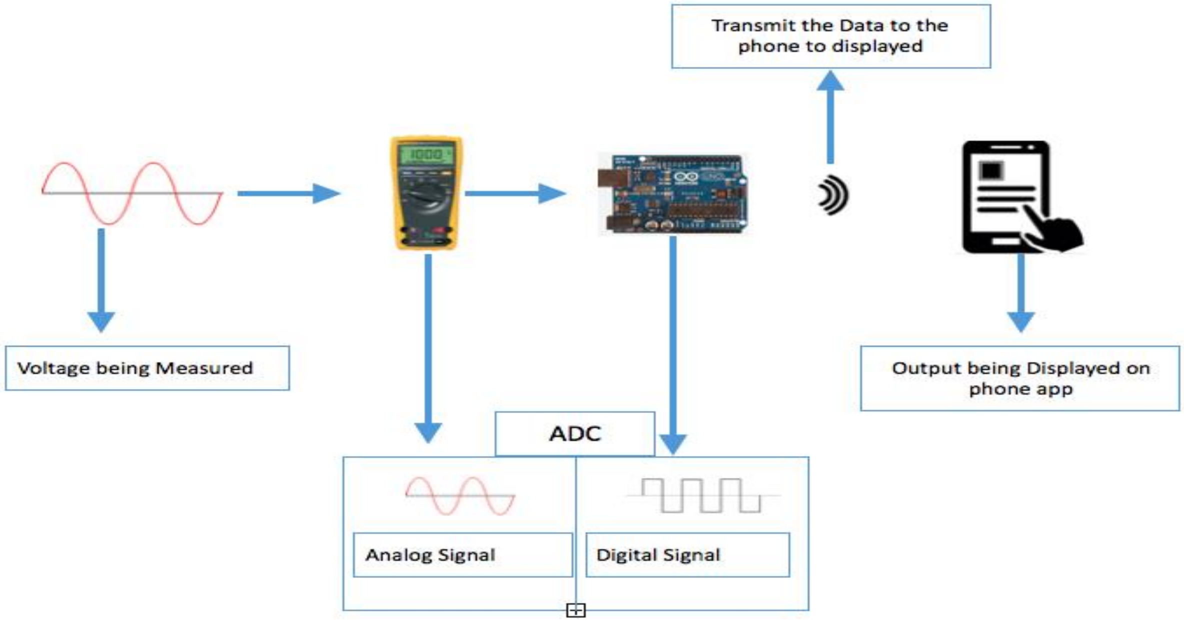


Figure 5.3: Conceptual Flow of the Program

Section 2: In this section we will go more in depth and talk about the code design for the microcontroller. The algorithm design is the piece that connect everything in our project. It is the network that bridges the electric circuitry, the Arduino, and the android application to make a fully functional wireless android application.

- **Power On :** This is where everything begins, we power the circuit and the Arduino, as soon as the system is running, we should get an indication that it is **"ON"** when microcontroller starts blinking. When the system does not blink, we should recheck everything on the board. One way of doing this is to turn ON/OFF to reboot the Arduino MCU. The reason why we are verifying that the MCU is on is so that the code is running smoothly on a fully functional and connected integrated circuit.
- **Declare Variables and set up Flag** - This is inside the program the code, it is in the first part of the code before we begin anything. We declare the variables we are going to need and use. These include the following: vmax, vmin,vrms, vin,vout etc. We are going to include flags in our code, they are meant to raise signal when an event occurs, for example suppose we measuring voltage but we are reading more than one input and all of them are greater than 0, this will activate or raise a flag notifying us to remeasure the voltages to make sure we are performing the correct tasks. Flags are meant to check if something is wrong without crushing the code. they are like a test condition and also used to store the state of the program.
- **AC/DC :** This is the part of the code where we check whether measuring DC or AC. Basically we are going to have the program run infinitely and check to

see which mode we are in. Once we have determine which mode we are working with, Read in the 4 analog inputs.. We check to see if more than more two analog readings have measurements of more greater than 0. This is where we activate the red flags if at least to analog input are greater than zero. Then we have to reread again until we get the correct measurement.

- **Voltage range ;** after we have checked the correct input voltage readings, the next step is to voltage the voltage ranges that the voltmeter is measuring. These ranges have not yet been determined. basically the circuit will control the voltages ranges then feed them to the program. The program will look at the received the input voltage and send it to the correct switch statement. The ranges are stored in arrays in the Arduino memory which faces the ADC. This is when sample the analog signal to digital in order for it to be stored, we can only store the signal if it is digital format which is a sequence of bits. We have not yet decided the mechanism of how we are going to achieve that . we are still in the process of determining where exactly we are going to sample the signal. but for now we assume, we are going to sample before we store. We are going to sample according the Nyquist sampling formula because we have to get the most accurate replication of the voltage signal in digital form.
- Calculation ; This is where we do calculation of everything we are going to need and display concerning the measurement of voltage. We are going to use the resolution formula to convert the voltage to the value being used. We are going to calculate the min, max, and averages and use take those and display them on the LCD on the board. After they have been displayed on the LCD we then Transmit the calculation we did on the Arduino board to the android smartphone through Bluetooth. once on the data is on the phone it get taken care of by the software on the Bluetooth.

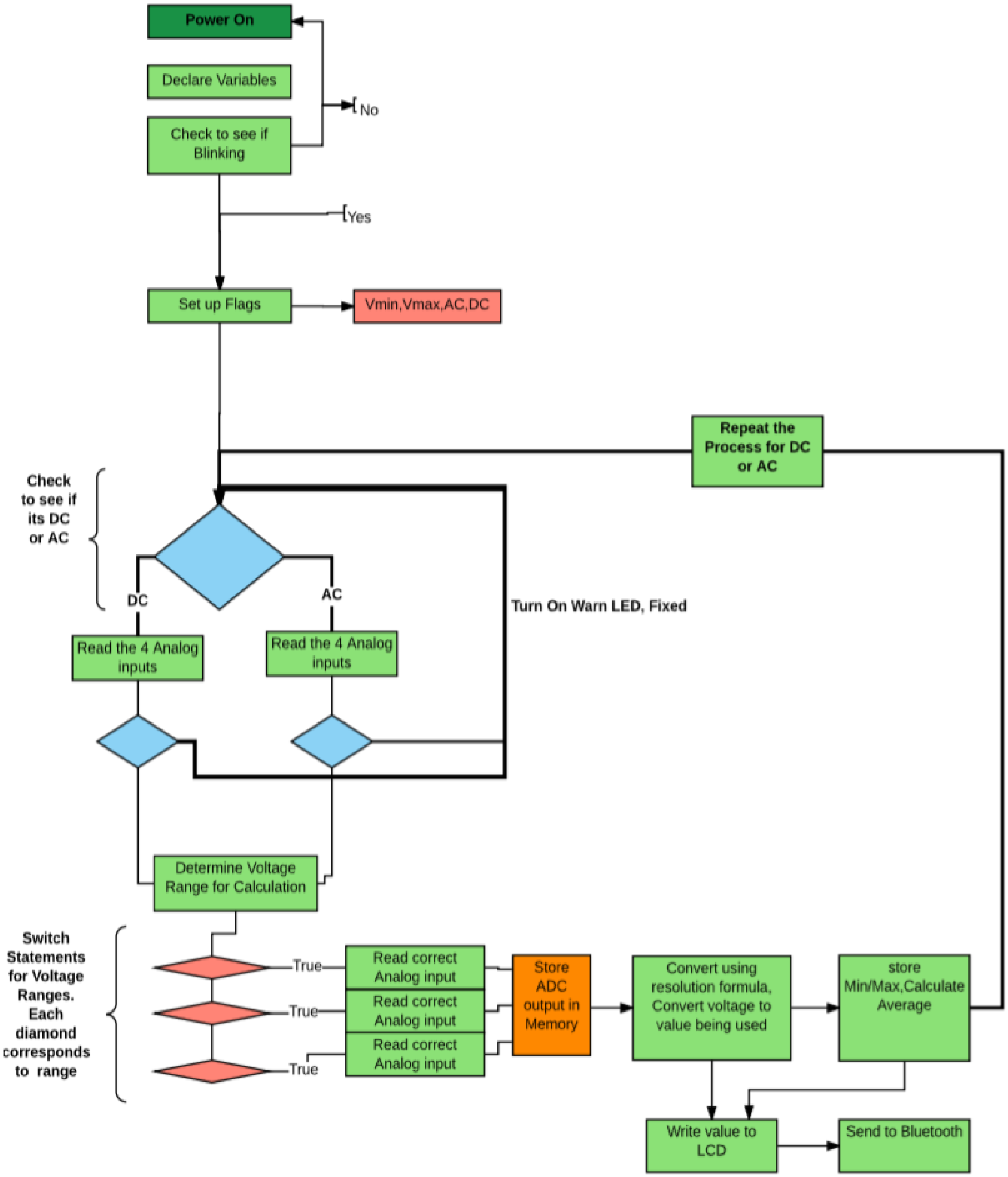
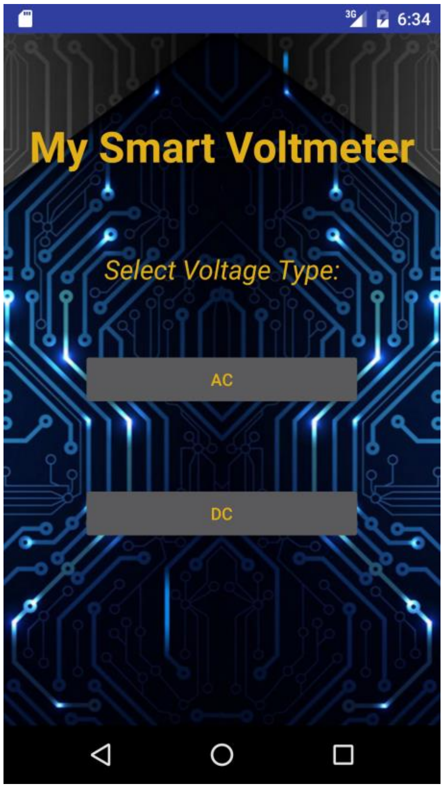


Figure 5.4: Code Design FlowChart

5.6 Summary of Application and Microcontroller Design

The Android application will be primarily a monitoring application mimicking the behaviors of an oscilloscope. The application will receive information transmitted through the Bluetooth® module of the voltmeter and adjust its data output accordingly. The options of data to be output will be the current type of the measurement wanted such as alternating current (AC) or direct current (DC), the measurement the values are taken at whether they are millivolts, volts, etc. and the actual values measured from the voltmeter which can be plotted on a graph for AC measurement or shown numerically for the DC measurement. The android application will be able to function on most of the android software versions in use today from the API 19, Kit Kat 4.4, and upwards. This application will also reach all requirements to be published as an official application on Google's Play Store. The application designing process, much like most aspects of this project has required fluidity in terms of final design and overall capabilities. The application has changed quite a bit since the original creation to combat the adapting nature of the project. The final design of the home screen of the application can be seen to the right, only allowing the user to choose between AC or DC voltage. If AC voltage is chosen, a graphical representation of the voltage will be given by default, however the user can choose to display that or the numerical representation by using the radio buttons marking the distinct screens. The numerical representation will look similar to that of the DC measurement, showing the current value being measured while also keeping track of potentially crucial information to the user such as the minimum voltage, maximum voltage, average voltage, etc. In the DC measurement ideally the voltages will not fluctuate rapidly in order to maintain accurate results. There will not be a graphical representation option for the DC measurement as the voltage should remain constant or very close to constant. Forcing a graphical representation in order to present a straight line graphed at a specific value could require more resources such as processing on both the voltmeter and mobile phone hardware which is detrimental to the overall operations of the system in place. Most of the data required for data representation should sent from the actual voltmeter hardware such as degree of voltage measured and potentially what measurement is being used such as AC or DC modes. The application will simply interpret the data sent over the Bluetooth® transmission and decode that data in a user friendly manner.



Basically the code for microcontrollers is designed to handle the analog input voltage that is being read into the integrated circuit. What the code does is to first check to see if the arduino is properly working by looking. Then we declare the variables that need to initialized like the baud rate, pinmode and such. when that is done. we have while loop that checks to see if it is DC or AC , depending on which path it takes, from there it reads in the voltage ranges then converts them to digital signal which are then taken for calculation of the values we want. These values are displayed on an LCD then transmitted to a bluetooth.

6 Prototype Construction and Schematics

In this section, we will cover the plan for breadboard testing and how this will be translated to the PCB. The schematics for each step will be presented and we will briefly discuss the plans for PCB layout. We will finish the section by selecting a vendor that will construct our design. In section 7, we will outline the specific part testing and results of each test and cover the strategic test procedures for the breadboard and PCB.

6.1 Breadboard Prototype Construction Plan

Once each of the components have been individually tested to ensure each is operating as expected, we will begin implementing the design using a breadboard and the Arduino IDE. The full and final design schematic is shown in Figure 6.6. This schematic shows all of the components and features that we plan to implement. For the sake of breadboard testing, we will begin prototype construction by assembling the microcontroller and supportive hardware components as shown by the schematic in Figure 6.1 below.

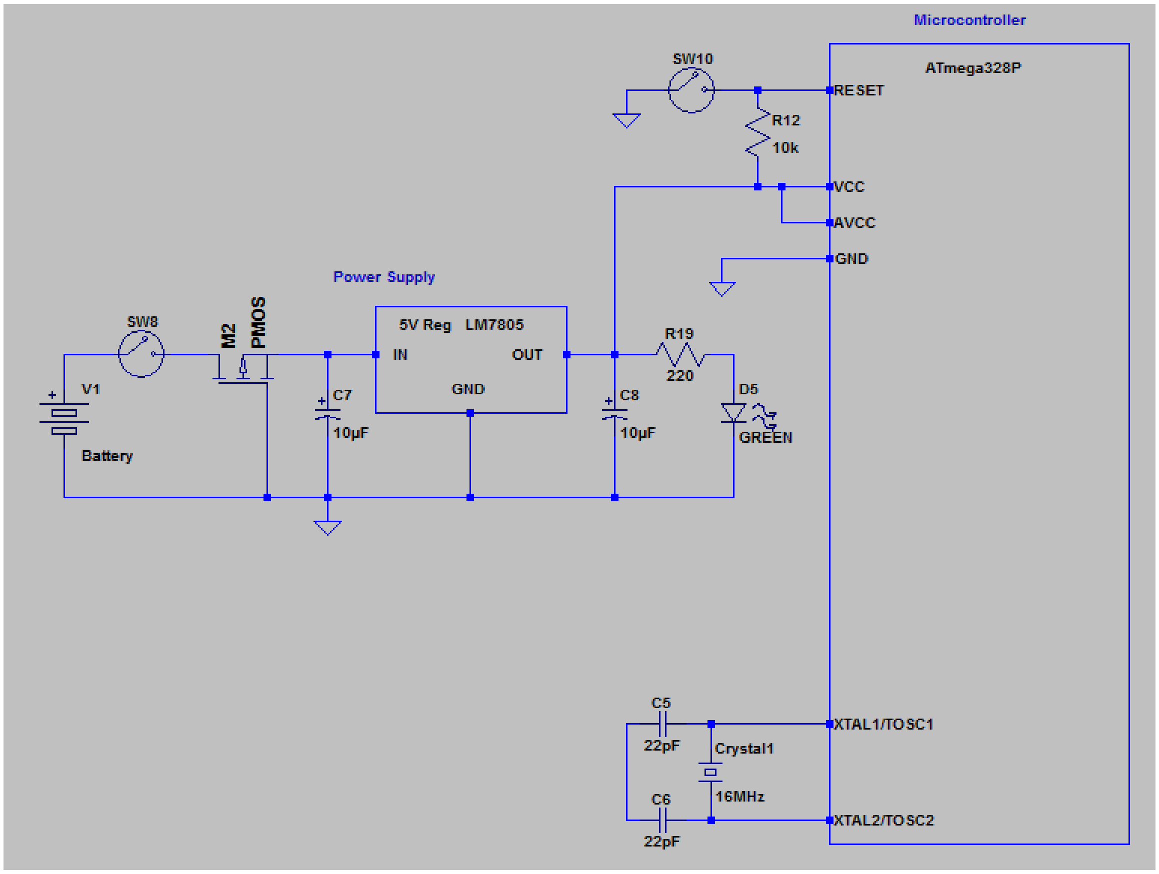


Figure 6.1: Prototype Construction - Step One

Although the power indication LED is not necessary, we will implement it in this portion of the design as it will be a useful indication that power has been connected correctly. Once the breadboard connections have been made, we will conduct the various tests outlined in section 7.4. These tests will ensure that we have full operation of the microcontroller and all features are working such as serial communication and programmability. Once all tests have been completed for the standalone microcontroller, we will implement the entirety of the input circuitry as shown in Figure 6.2 below. In this setup, we will be testing the analog inputs by measuring a wide range of AC and DC voltages while recording the voltage drop on R2 through R7. Because the AC/DC selector switch will not be installed at this step, we will have to manually switch between AC and DC modes within software and ensure each mode is fully operational (Excluding wireless transmission). We will compare these values and adjust the potentiometers to create the exact resistance values calculated in section 3.5.7. To complete this test, we will need a power supply that can generate large voltages such as the ones found in the engineering labs at UCF. Once we have found the correct resistance values, we will write a raw microcontroller code to read the analog inputs as we test all ranges and feed the ADC outputs and calculated voltage values back to the Arduino IDE through serial communication. Once we are confident that the ADC is operating correctly for all ranges, we will connect the LCD and attempt to write and display the calculated values.

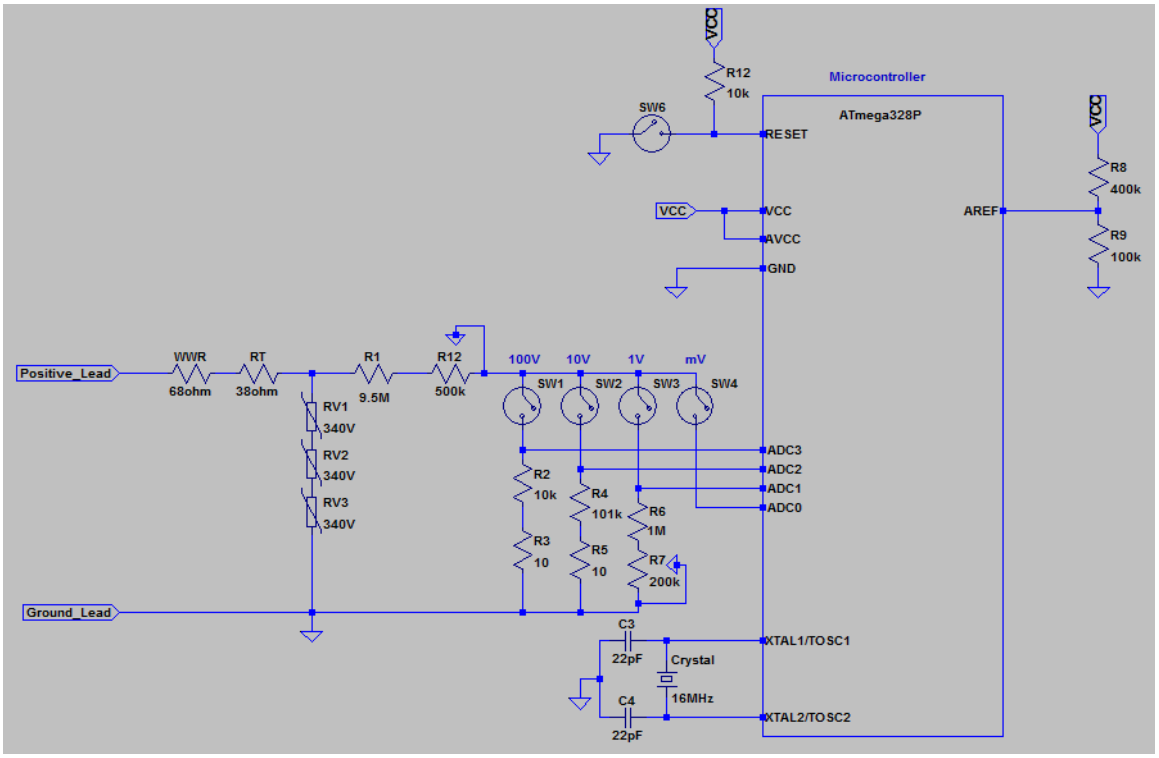


Figure 6.2: Prototype Construction - Step Two

*Note: Power supply not pictured

The LCD connection shown in Figure 6.3 below will be made using the same breadboard and connections shown in the previous steps. Once the LCD has been connected, we will workshop the previous code to include Arduino's Liquid

Crystal library and write the measured voltage values to each of the lines on the LCD and ensure the values match the measured values and voltages.

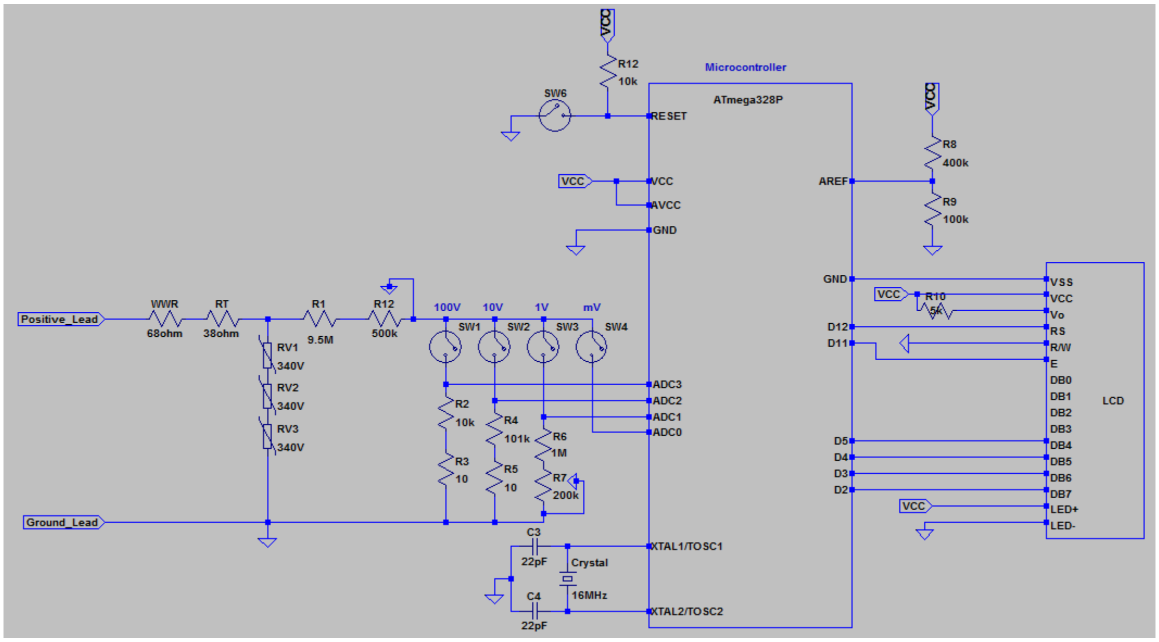


Figure 6.3: Prototype Construction - Step Three
*Note: Power supply not pictured

As the LCD, along with the pull-up resistors and AREF voltage divider are connected to the microcontroller's ground, we know that this will affect our input voltage divider and thus measured values. That being said, we will repeat the measurement tests completed in step 2 (Specific test procedure is outlined in section 7.4). By slowly adjusting the potentiometers and subsequent variables in the microcontroller code, we will continuously measure and revise our design through the trim pots to match the accuracy achieved in the previous step. Our margin of error at this point is to be within 1% accuracy of the digital multimeter used in the lab. Once we have achieved this goal and are able to display each of the calculated voltages, we will work to optimize the code before attempting to implement the Bluetooth® module and the remaining switches and LEDs.

The next step in the prototype construction will be to include the AC/DC selector switch, AC/DC indicator LEDs, and range selection warning LED as shown in Figure 6.4 below. We will then include the necessary code modifications to incorporate these new variables. The inclusion of these components will inherently contribute to the overall current draw and noise presented on the pins of the microcontroller. Knowing this, we will continue to repeat the thorough measurement tests from step 2 and make adjustments accordingly.

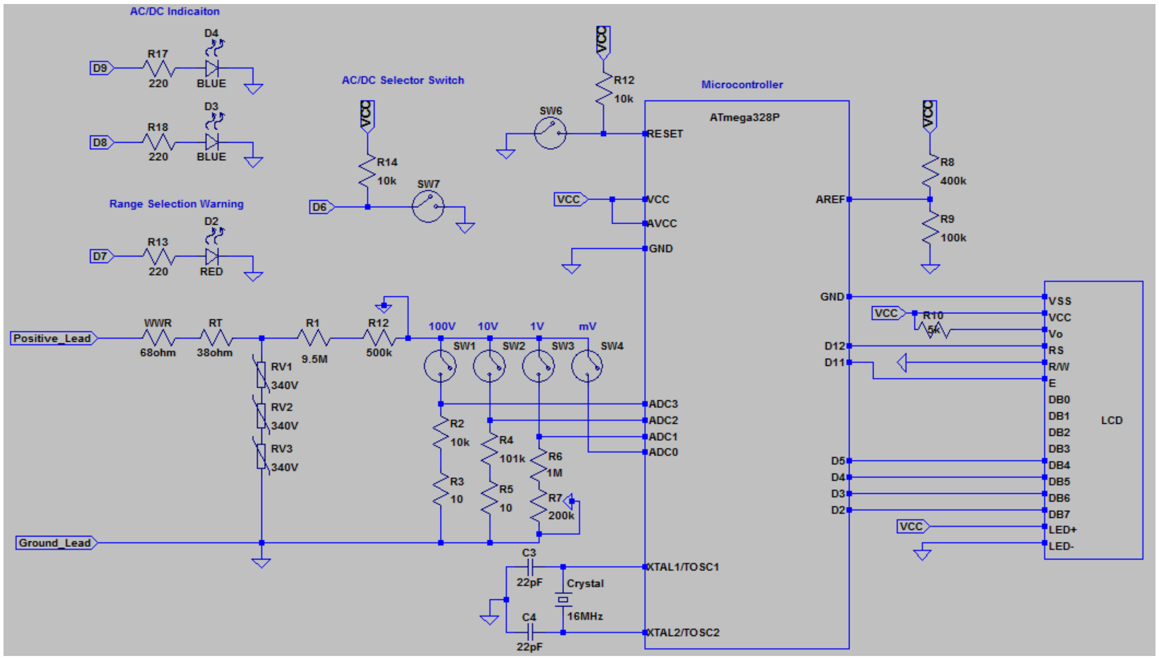


Figure 6.4: Prototype Construction - Step Four
*Note: Power supply not pictured

After we have made the above connections and are satisfied with all features tested thus far, we can connect the Bluetooth® module as shown in Figure 6.5 below. After making the necessary connections, we will attempt to send the voltage values to the S2 Terminal application found in the Google Play Store. In this application, we will record the hex values received and ensure they match the values sent from the microcontroller and thus our measurement values.

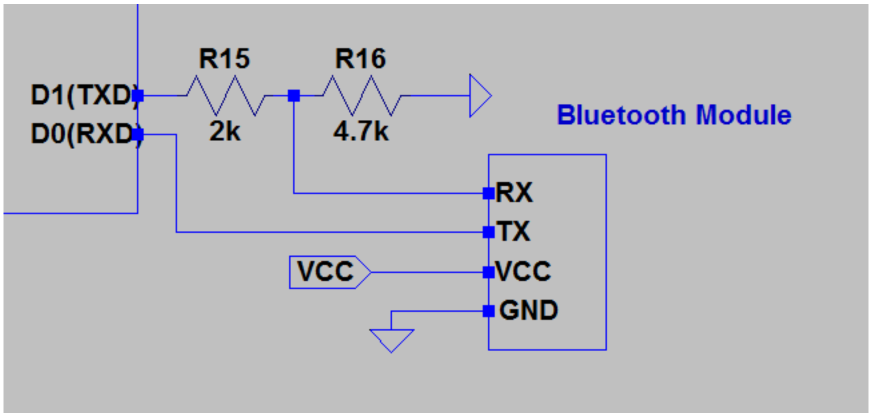


Figure 6.5: Prototype Construction - Step Five
*Note: The remaining components are not shown, but will be connected as in the previous steps

The final step in the prototype construction will be to finalize the smartphone application and make a connection to our Bluetooth® module. As presented in the following section, we will conduct a series of tests to ensure the full functionality of the application. If all goes as planned, we will follow a strict testing regimen that covers all features of our product and all possible scenarios of measurement. Once we are satisfied with the results of these tests and our prototype is fully functional, we will begin laying the plans for PCB and package construction.

At this point in the prototype construction, we should still be within the 1% margin of error for our measurement values. Should we not be able to meet this requirement, we could potentially need to make changes to the input resistance values by implementing various potentiometers. Once this requirement has been met, we will begin optimizing the microcontroller code and application parameters to meet our final requirement specification of 0.1% accuracy.

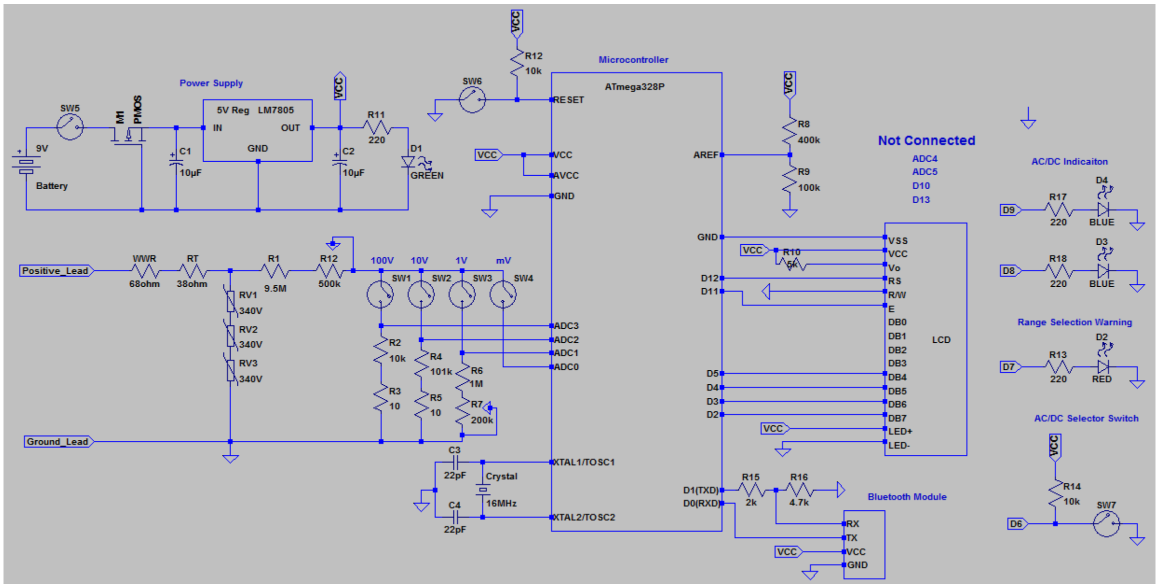


Figure 6.6: Final Schematic

6.2 PCB Vendor and Assembly

After much deliberation pertaining to the various PCB assembly companies available to order from, the PCB vendor will be ExpressPCB.com. We decided to go with this vendor because they provide free software to design the PCB and quickly upload this design to their manufacturing division. This means the PCB design process should be completed in a relatively timely fashion and multiple PCBs will be ordered in order to combat any potential defects in specific PCBs. This vendor was also recommended by previous engineering students.

For the transition from breadboard to PCB, we will use multiple resources when designing the layout for optimal design. Once the PCB has been designed, we will connect and test the components in the same order as done for the prototype breadboard construction. We plan to use high quality solder and complete all construction steps in a large room using extractor fans for ventilation purposes. The construction steps for the breadboard prototype will be the same steps taken when constructing the PCB. Additionally, at each step of the PCB assembly, we will follow the strict testing procedures completed in the prototype testing so that we may narrow down any issues quickly.

Because the PCB will be designed in a compact fashion, the measurement components and microcontroller will be in close proximity. This inherently presents multiple safety concerns for our product which must be addressed when designing the layout. As we mentioned in section 3.4, our product will have to take all possible hazards into consideration throughout the design process. Through our extensive research we selected the electrical components to protect both the meter and user from these hazards, but the safety measures cannot stop at the component level. For added protection, we will design our PCB to include isolation slots between high-voltage components and under the varistors as well. Although unlikely, there is the possibility for large voltages to arc over the electrical components in our meter which can compromise their functions which could present dangerous and unsafe conditions. We have already taken this potential hazard into consideration when we chose to use three lower voltage varistors instead of a single large one. By expanding the spacing between these components with the addition of isolation slots, the risk of high-voltage arc is greatly reduced.

Other factors to consider when designing our PCB layout are reducing electromagnetic interference, reducing noise on the sensitive I/O pins, and possible undesired electrical effects caused from high voltage inputs. When measuring large voltages, our product could potentially experience RF coupling between components and wire traces if not properly designed. This potential effect is generally caused through inductance on high current traces which produce voltage drops that radiate throughout the PCB. Though this will be kept in mind when designing the layout, measures such as the large nominal input impedance that will limit the current draw into the meter will help in preventing this undesired effect. As we have spent the majority of our project researching and designing the meter for correct operation, we have not devoted much time into the research for the PCB design. Though we have a general idea of the parameters which must be considered when designing the layout, we still have much to learn in Senior Design 2.

7 Prototype Testing Plan

7.1 Hardware Test Environment

Overall the voltmeter designed will be tested vigorously in order to assure not only precision and usability, but also safety due to the possibility of high voltage and/or current measurement by the user. Testing the voltmeter will include the creation of various test circuits with predetermined voltages which are measured using existing and calibrated multimeters. These measurements will be compared to the digital multimeter in order to ensure quality and precision is maintained consistently. 10 different predetermined circuits will be tested on both the digital multimeter and the existing multimeter and compared. Safety is also a major concern due to the nature of the use of the voltmeter. This meter may be coming into contact with high voltage circuitry and in an instance where a circuit is compromised or faulty which could be potentially dangerous to a user, the meter must be able to withstand disabled operation and the transfer of electricity to the unit via the safety components in the circuitry of the meter.

This will be tested by introducing a voltage greater than the rated voltage of the digital voltmeter to ensure proper functionality of the safety precautionary systems put in place. The meter will be first tested in controlled laboratory environments in which temperature, humidity, and other external conditions are minimized in order to ensure proper functionality. Based on the portable nature of the voltmeter, after successfully passing tests conducted in the controlled environment the product will be given similar circuitry to measure while in areas of its likely use, such as outdoors, constrained environments, etc.

7.2 Individual Component Testing

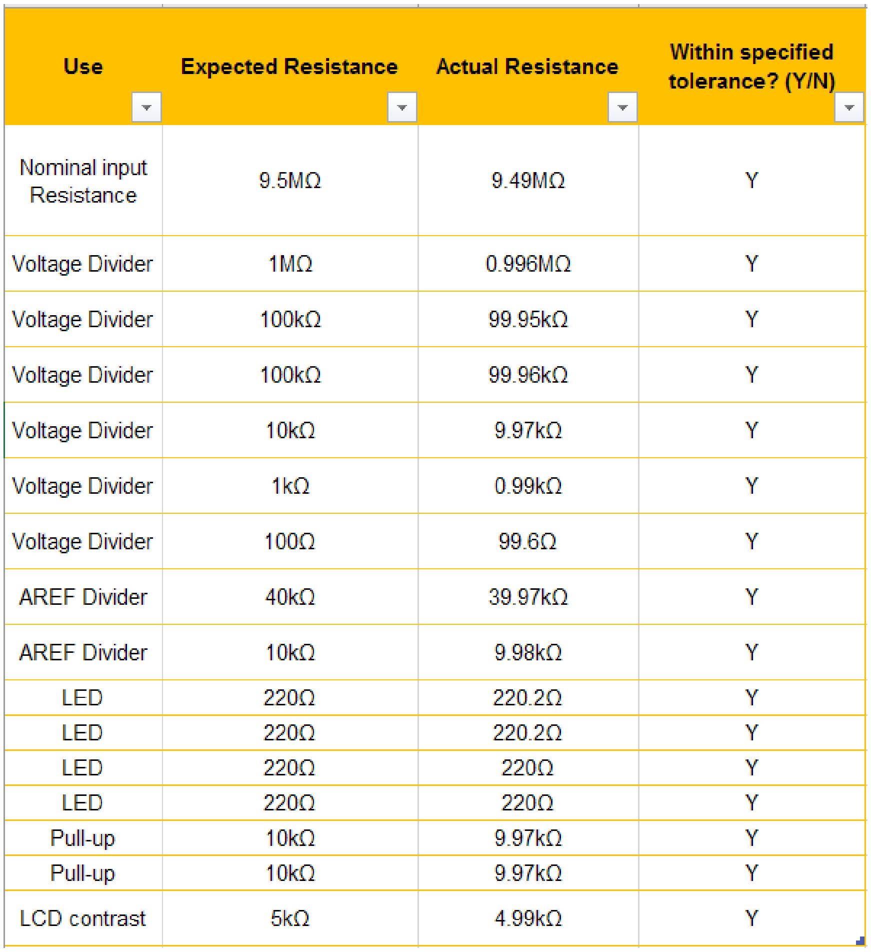
To ensure that our meter would operate correctly, we had to be sure that all components used were in working order. In this section, we will describe how each part was tested and the results of each test. These components will be tested in the order shown below.

1) Resistors, capacitors, test leads and switches
2) Safety Components
3) Battery and voltage regulator
4) Diodes
5) MCU and crystal
6) LCD and Bluetooth® Module

By testing the components in this order, we can ensure that each component is operating efficiently. Additionally, we will be able to determine any possible issues quickly as each step uses the components preceding them.

7.2.1 Resistors, capacitors, test leads and switches

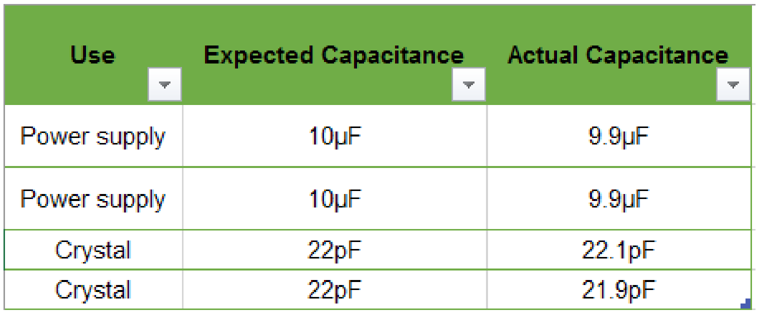
The first components we decided to test were the resistors to be used in the final design. These components are crucial to the success of our multimeter, so we needed to ensure that the actual resistances were within the tolerance ranges for each specific part. The list of all resistors to be used in the final construction are listed in Table 7.1 below.



Use	Expected Resistance	Actual Resistance	Within specified tolerance? (Y/N)
Nominal input Resistance	9.5MΩ	9.49MΩ	Y
Voltage Divider	1MΩ	0.996MΩ	Y
Voltage Divider	100kΩ	99.95kΩ	Y
Voltage Divider	100kΩ	99.96kΩ	Y
Voltage Divider	10kΩ	9.97kΩ	Y
Voltage Divider	1kΩ	0.99kΩ	Y
Voltage Divider	100Ω	99.6Ω	Y
AREF Divider	40kΩ	39.97kΩ	Y
AREF Divider	10kΩ	9.98kΩ	Y
LED	220Ω	220.2Ω	Y
LED	220Ω	220.2Ω	Y
LED	220Ω	220Ω	Y
LED	220Ω	220Ω	Y
Pull-up	10kΩ	9.97kΩ	Y
Pull-up	10kΩ	9.97kΩ	Y
LCD contrast	5kΩ	4.99kΩ	Y

Table 7.1: Resistor Testing

Now that we have confirmed the resistors that will be used for the meter are within their designed tolerances, we can move on to testing the capacitors. Similarly to the above process, we tested each of the capacitors using the capacitance mode of our store-bought meter. The test results can be seen in Table 7.2 below.



Use	Expected Capacitance	Actual Capacitance
Power supply	10µF	9.9µF
Power supply	10µF	9.9µF
Crystal	22pF	22.1pF
Crystal	22pF	21.9pF

Table 7.2: Capacitor Values

As you can see from the table above, the capacitors are operating as expected. The next section of testing involves continuity tests for the test leads and switches. We began by first testing the resistance of each test lead from the probe to the banana jack. Both of the probes had 2Ω resistance which is satisfactory for our design. We then performed the same test on the range-selector DIP switch, the power supply switch, and both push buttons. Each of the contacts were 5Ω or lower which indicates all switches are operating correctly.

7.2.2 Safety Components

To ensure all of the safety components were in working order, we had to perform a variety of tests for each application. Unfortunately, some of the components require large voltages and currents to test their functionality. These tests would require high-voltage power supplies and would present multiple safety hazards. Because this is only a proof of concept, we will have to trust the manufacturer's specifications and assume their values are correct. However when the time comes to build the full meter and test its functionality, we will need a power supply that can produce these large voltages. If we are unable to obtain a power supply, we can construct a makeshift supply that amplifies and rectifies 120VAC from a wall outlet. The AC signal would first pass through a 100µF capacitor which will hold the peak of the 120V input which is 170V. The first diode prevents the capacitor from discharging to ground which allows the voltage to be constant at 170V. Similarly, on each positive cycle of the signal, the output capacitor is charged to the peak voltage plus the first capacitor's voltage which produces a 340V DC output. The schematic for this circuit is shown in Figure 7.1 below.

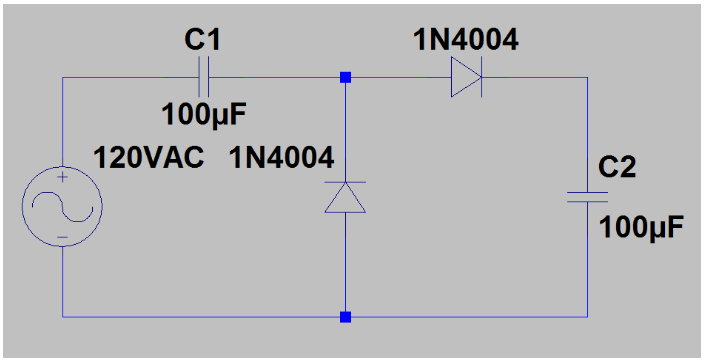


Figure 7.1: Varistor Testing Power Supply

As mentioned in section 3.4, we chose to place three 340V varistors in series to give us a maximum clamping voltage of 1020V. We can use this power supply to produce 340V and test each of the three varistors. By placing the varistor in parallel with the capacitor, the current should be shunted through the varistor which we can measure with a store-bought ammeter.

Similarly to the varistor test, in order to test the high-wattage resistor which would absorb the large amounts of energy produced by the varistors shunting the current back through the negative lead, we require a power supply that can produce a large current. While we have access to a power supply which can potentially produce this large of a current, the safety hazards again take precedent and we will not be able to confirm the actual power dissipation parameters of this component. On the other hand, when it comes to the PTC thermistor, we can connect this component to a power supply, raise the current passing through the component and measure the resistance across the leads. While this is safe in theory, in the event that the component is not to specification or is possibly shorted, there is a chance of shorting our power supply and damaging the equipment. Fortunately, we can perform a makeshift test by connecting the thermistor to a store-bought meter and measure the change in resistance as we hold the component increasingly closer to an open flame. We performed this test and noted that the resistance at room temperature was 38.4Ω. After holding the thermistor close to an open flame for a few seconds, we saw that the resistance increased to over 200Ω. This confirmed that the component was operating as expected, although we cannot determine the limits of operation.

7.2.3 Battery and Voltage Regulator

When testing the battery, all that we needed to do is connect the probes of our store-bought meter across the positive and negative terminals to measure the voltage. Doing so, we measure our battery to actually be 10.02V out of the package. Although this is larger than expected, our voltage regulator should be able to handle input voltage from 7-16V. This led us to the testing of our voltage regulator. Using a breadboard, our two 10µF decoupling capacitors and our 9V battery, we connected the 7805 voltage regulator in the circuit shown in Figure 7.2 below.

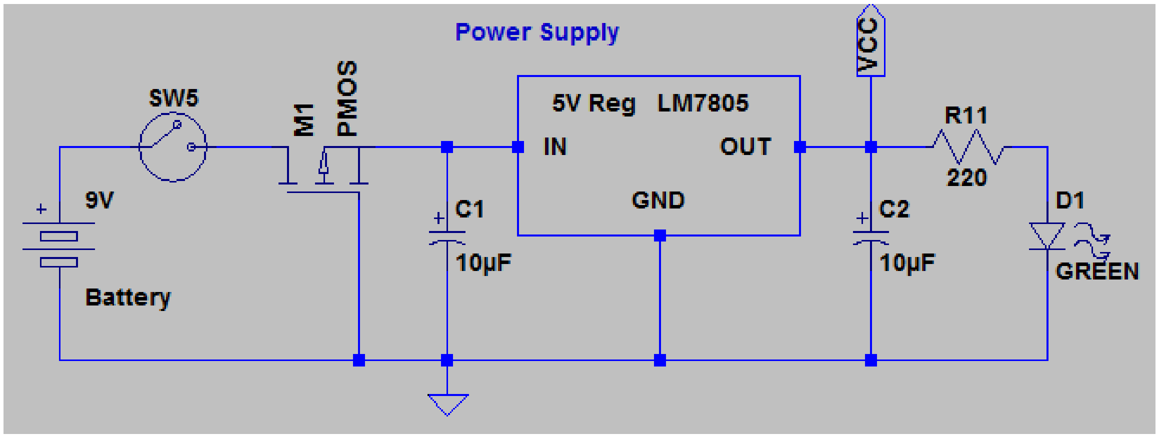


Figure 7.2: Power Supply Circuit

We measured the voltage on the output of the regulator to be 5.02V which is sufficient for our design. To test the limits of this component, we connected a power supply to the input and varied the voltage from 12V down to 4V. At 4.9V, the output of the regulator dropped below 5V and followed the input. This confirmed that the component was operating as expected and would work for our device.

7.2.4 Diodes

The next step in our individual parts testing was to test the diodes that we ordered for our project. As outlined in our final design, we eliminated the need for a diode for AC rectification as proposed in section 5. The diode that would have been used for power supply protection was replaced with a P-channel MOSFET. Although we still plan to use this component, we wanted to confirm the functionality of the 1N5818 Schottky diode that we would use as an alternative. To test the rectifying diode, we connected a power supply to the circuit as shown in Figure 7.3 below. Two tests will be performed; the first will be a DC sweep from -12V to 12V with R1= 1kΩ and the second will be to change the value of R1 from 100Ω to 10kΩ at a constant Vin. The first test will be determine if the diode is biasing correctly, and the second will vary the amount of current passing through the circuit and will show us the range of voltage drop depending on the load. We will use a digital multimeter to measure the voltage at Vout. From -12V to 0V there should be no current flowing through the diode as it will not be forward biased. Above zero volts to 12V the diode should be forward biased and allow current to flow through the circuit. We ran a simulation of the first test and got the result shown in Figure 7.4.

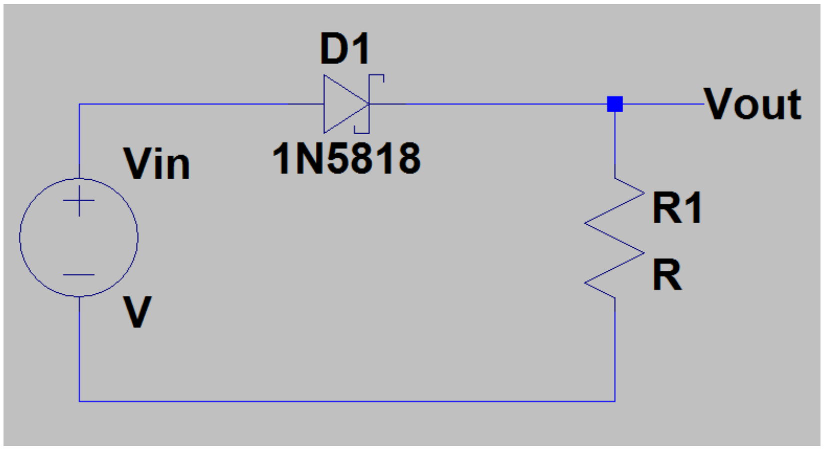


Figure 7.3: Diode Test Circuit

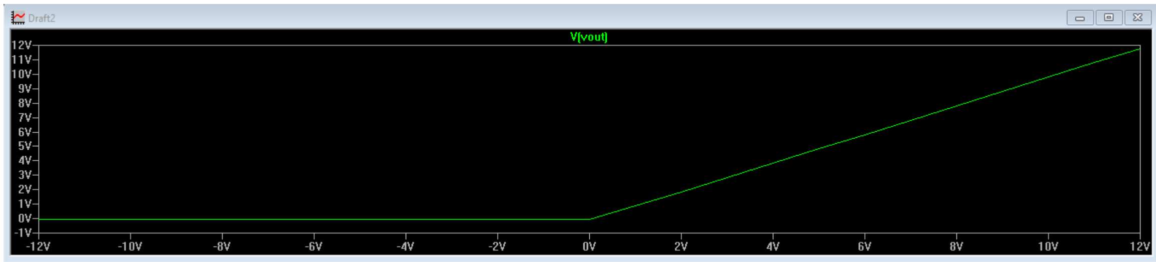


Figure 7.4: Diode Test Simulation

When the voltage rises above the threshold voltage, we expect to see the voltage at Vout follow the input, only slightly lower as there will be a drop across the diode. For the second test, we should see the voltage drop across the diode vary slightly as more current is passed through the diode. We will take the appropriate measurements for each test and compare the results with the datasheet.

The final diodes that needed to be tested were the indication and warning LEDs. We connected a 220Ω resistor to the 5V regulated output of our power supply and inserted the green LED. We confirmed that the LED was shining brightly and not getting too hot. We performed the same test with the red and blue LEDs and obtained the same result. This confirms both our design and that the LEDs are functioning correctly.

7.2.5 MCU and Crystal

Now that we had an operating power supply, we had to test the remaining components necessary to operate the microcontroller chip. We began by placing the ATmega328P into the compatible breadboard socket and placed it on the breadboard. Using our 5V regulated power supply, we made all of the necessary connections to VCC and GND pins on the IC. We then connected the 16MHz crystal and 22pF capacitors as shown in Figure 7.5.

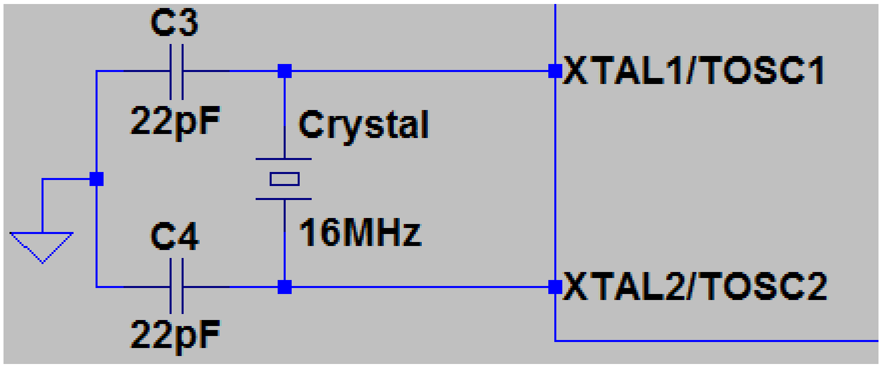


Figure 7.5: Crystal Oscillator Connection

Lastly, we connected the pullup resistor from 5V to the reset pin of the microcontroller. In order to confirm the operation of these devices, we had to connect a laptop with the Arduino IDE to the circuitry. To do so, we connected our USB to serial breakout board to the appropriate pins of the IC, power and ground. We opened the Arduino IDE and confirmed that the program recognized the device by writing a simple code to blink an LED on digital pin 13. We connected a 220Ω resistor in series with a red LED to ground and confirmed that the LED was blinking. This test proved that both the microcontroller and crystal were operating as expected.

In order to further test the microcontroller, we connected a 10kΩ potentiometer from 5V to the input of AIN0 which is an analog input to the ADC. We then established serial communication to the board and through Arduino's serial monitor, we were able to feed in the output value of the ADC as we varied the potentiometer. We confirmed that the output values of the ADC went from 0 to 1023 which was expected as the preset reference voltage of the ADC is 5V.

7.2.6 LCD and Bluetooth® Module

In order to test the the LCD and Bluetooth® module, we decided to first test their functionality using the Arduino Uno we used in our initial proof of concept. Because we could still have potential issues with our standalone microcontroller, we thought it best to ensure the individual components were operating correctly on an Arduino development board. We connected the LCD as shown in Figure 7.6. with a 10kΩ potentiometer between Vcc and Vo of the LCD which controls the contrast. We then used the Arduino IDE to write 10 different example programs provided in Arduino's Liquid Crystal library. These example programs make use of each function available in the library that can be used to control all aspects of an LCD. After performing these example codes, we were confident that our LCD was in perfect working order.

The final hardware we have to test is the Bluetooth® module. In order to do this, we have to first connect the module to the appropriate pins of the Arduino. The transmit (TX) pin of the Arduino will be connected to the receive (RX) pin of the Bluetooth® module and vice versa. Because the Bluetooth® module operates on

3.3V logic levels, we have to make a simple voltage divider that will drop the 5V signals being output from the MCU's TX pin to 3.3V. The resistors we will use are 2kΩ for R_1 and 4.7kΩ for R_2. The remaining connections are simply power and ground. We then have to create a simple code that sends data for the Bluetooth® module to send to a smartphone application. Android has several free applications that allow users to connect to Bluetooth® devices such as this. We will be using an application called S2 Terminal for Bluetooth® as our application has not been fully developed yet. The code we will write will send a variety of data to the application, and the application will send the data back to the Arduino. Upon successful transfer of data in both directions, we will have confirmed that our Bluetooth® module is working correctly.

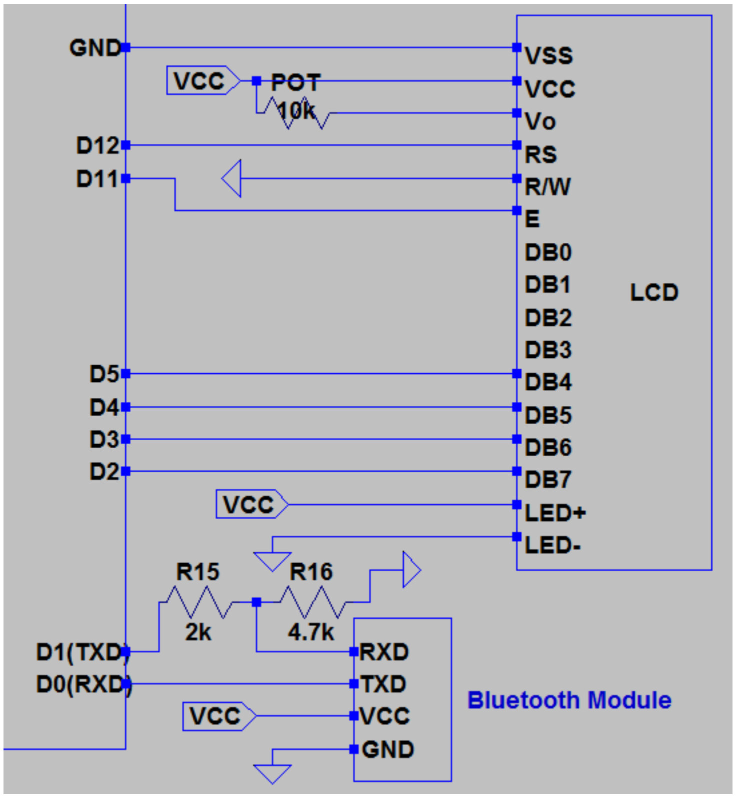


Figure 7.6: LCD and Bluetooth® Implementation

7.3 Breadboard and PCB Test Procedure

In this section, we will outline the specific test procedure we plan to conduct that will ensure full functionality of the product. Upon the success of each individual component tests, we will construct the breadboard prototype in the steps listed in section 6. For each of the steps, the following tests listed below will be performed in order once construction has been completed. Table 7.3 lists all of the necessary equipment that will be used in the following test procedures.

Equipment	Use	Item Indicator
High-voltage power supply	Producing DC voltage for measurement	a
Signal generator	Producing AC signals for measurement	b
Digital Multimeter	Continuity tests and comparison control	c
Oscilloscope	Measuring AC signals for comparison	d
Laptop + Arduino IDE	Programming and serial communication	e
USB to Serial breakout board	Programming and serial communication	f

Table 7.3: Test Procedure Equipment

Step 1: Standalone Microcontroller

1. Visual inspection and continuity
    a. Ensure all connections in Figure 6.1 have been made
    b. Using item [c], ensure all point-to-point shorts are <5Ω
2. Power up and programs
    a. Power the device and connect [e] to MCU through [f]
    b. Connect LED and 220Ω to digital I/O D0 of MCU
        i. Write program to blink LED with a delay
        ii. Confirm correct operation
        iii. Repeat for each remaining digital I/O
        iv. Remove resistor and LED
    c. Connect pushbutton, 10kΩ pull-up resistor to digital I/O D0 of MCU
        i. Write example programs below and confirm expected values
            1. Button
            2. Debounce
            3. StateChangeDetection
    d. Connect 10kΩ potentiometer and 10kΩ resistor to analog input A0 between VCC and ground
        i. Write example programs below and confirm expected values
            1. AnalogReadSerial
            2. ReadAnalogVoltage
        ii. Remove potentiometer and resistor

Step 2: Measurement Circuitry

1. Visual inspection and continuity
    a. Ensure all connections in Figure 6.2 have been made

   - b. Using item [c], ensure all point-to-point shorts are <5Ω
   - c. Using item [c], adjust R12 to create ~10MΩ between range switch and positive lead
   - d. Using item [c] measure and record resistances of R2-R7
     - i. Make necessary adjustments to match values in Figure 3.1

2. DC measurement
   - a. Disconnect all jumper wires from MCU analog inputs
   - b. Select mV range using the Range Selector Switch
   - c. Using item [a], generate a 0V DC signal, connect output to test leads of meter
   - d. Using item [c], measure and record voltage at the end of the jumper wire removed from A0 in step a
   - e. Sweep the voltage from 0-1V with 10mV increments and repeat steps c and d
   - f. Confirm recorded values match expected values and make adjustments accordingly
   - g. Select next range and repeat above steps, incrementing from 1-10V using 100mV increments
   - h. Select next range and repeat above steps, incrementing from 10-100V using 1V increments
   - i. Re-connect jumper wires to analog inputs
   - j. Write example program “AnalogReadSerial” to MCU
     - i. Repeat steps b through h while recording the output values of the ADC being displayed in the serial monitor of [e]
     - ii. Confirm for each range, the ADC output values follow the input from zero to 1023
3. AC measurement
   - a. Repeat steps a and b of the previous test
   - b. Using item [b], generate a 10mV signal at 1kHz and connect output to test leads
   - c. Using item [d], record the waveform at the end of the jumper wire removed from A0 in step a
   - d. Repeat steps e through i of the previous test
   - e. Write program to sample the ADC at maximum speed for ~0.2 seconds and print the values to the serial monitor of item [e]
   - f. Inspect the printed values and locate the positive and negative cycles of the input signal
   - g. Determine the number of cycles recorded and calculate the frequency of the signal
     - i. Confirm this value matches the 1kHz input
     - ii. If necessary, adjust the sampling frequency and length
   - h. Confirm peak values for each range have been detected
   - i. Repeat steps b through h at the following frequencies
     - i. 0.1 kHz
     - ii. 1 kHz
     - iii. 10 kHz

        iv. 100 kHz
        v. 1MHz

Step 3: LCD implementation

1. Visual inspection and continuity
    a. Ensure all connections in Figure 6.3 have been made
    b. Using item [c], ensure all point-to-point shorts are <5Ω
2. Repeat DC and AC measurement procedures to ensure the values match the ones recorded with the LCD disconnected in Step Three
3. Using item [e], write a program to calculate the current voltage (V), the average voltage (Vavg), the minimum voltage (Vmin) and the maximum voltage (Vmax) when a 10V DC signal is applied at the input
    a. Write the calculated values to the LCD
    b. Ensure values displayed match the values generated at the input
4. Using item [e], write a program to calculate the RMS voltage (Vrms), the peak voltage (Vpeak), and the frequency when a 10V AC signal at 1 kHz is applied to the test leads using item [b]
    a. Write the calculated values to the LCD
    b. Ensure values displayed match the values generated at the input

Step Four: Pushbutton and LED implementation

1. Visual inspection and continuity
    a. Ensure all connections in Figure 6.4 have been made
    b. Using item [c], ensure all point-to-point shorts are <5Ω
2. Combine the AC and DC measurement programs written in the previous section to include the AC/DC selector switch and indication LEDs
    a. The program should now switch between modes when the pushbutton is pressed (default to DC on powerup)
    b. Write the program to the MCU and repeat the AC and DC tests performed in Step Two
    c. Ensure recorded values match the previously recorded values within 1%

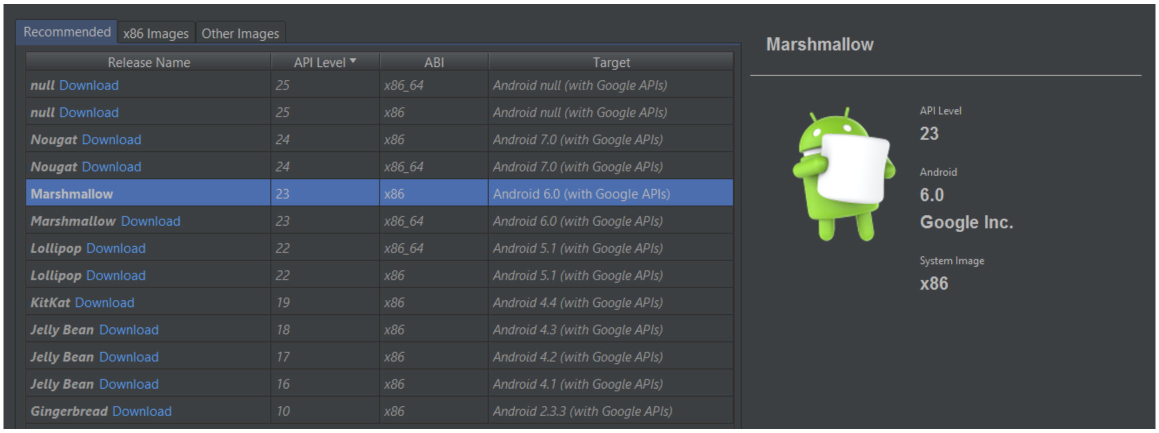
Step Five: Bluetooth® implementation

1. Visual inspection and continuity
    a. Ensure all connections in Figure 6.4 have been made
    b. Using item [c], ensure all point-to-point shorts are <5Ω
2. Establish connection between smartphone and Bluetooth® module using the S2 Terminal application
3. Modify the above program to incorporate the Bluetooth® module parameters
    a. Repeat steps 3 and 4 of “Step Three” and write the calculated values to the RX pin (connected to TX of MCU) of the module
    b. Ensure the values received in the S2 Terminal application match the transmitted values
    c. Additionally, ensure the values in the previous step match the voltages being applied at the input

7.4 Application Test Environment

The Android application will be tested using multiple environments both throughout and after the development process. This will be done using the same device as the development of the application in order to remain consistent with the development conditions for example Windows operating systems, Android Studio versions, etc. As previously stated, the application will only be available for the latest Android operating system, so mobile devices running Apple or Linux operating systems, or even versions of the Android operating system that are below the Android software release deployment 4.4 (Kit Kat) may not allow optimal functionality. The Android development platform, Android Studio, also includes a virtual Android operating system emulator which can be used to simulate application operations in a controlled environment on multiple Android operating system versions. This utility will be leveraged to ensure compatibility is maintained across all platforms and versions of the Android operating system listed as compatible with the software.

An image of the emulating utility is pictured to the below, presenting the amount of possible virtual testing conditions able to be simulated. Many more versions are available than the ones shown in the image, however not all will be required to test as the APIs extend past 19, which is the corresponding API to Android 4.4 Kit Kat. The newest version of Android, Android 7.0 (Nouget) API 24 will be tested to ensure functionality on the most up to date version of Android, and testing will commence down the list of APIs in decreasing order until API 19 is reached and confirmed functional.



Release Name	API Level	ABI	Target
null Download	25	x86_64	Android null (with Google APIs)
null Download	25	x86	Android null (with Google APIs)
Nougat Download	24	x86	Android 7.0 (with Google APIs)
Nougat Download	24	x86_64	Android 7.0 (with Google APIs)
Marshmallow	23	x86	Android 6.0 (with Google APIs)
Marshmallow Download	23	x86_64	Android 6.0 (with Google APIs)
Lollipop Download	22	x86_64	Android 5.1 (with Google APIs)
Lollipop Download	22	x86	Android 5.1 (with Google APIs)
KitKat Download	19	x86	Android 4.4 (with Google APIs)
Jelly Bean Download	18	x86	Android 4.3 (with Google APIs)
Jelly Bean Download	17	x86	Android 4.2 (with Google APIs)
Jelly Bean Download	16	x86	Android 4.1 (with Google APIs)
Gingerbread Download	10	x86	Android 2.3.3 (with Google APIs)

Marshmallow

API Level
23

Android
6.0
Google Inc.

System Image
x86

In addition to utilizing the Android Studio provided emulating software shown above, the application will also be downloaded onto multiple portable devices in order to test for functionality in physical portable devices and the performance of the application such as hardware and memory allocation, energy consumption while running application, and overall reactivity to application given commands in and out of the application. These portable devices will have varying Android versions as well as hardware specifications such as RAM and hard drive space in order to expose the application to various degrees of variables to test through. These portable devices including specifications will be listed after testing has begun.

7.5 Application Specific Testing

The digital voltmeter application involves many intricate programming techniques to create, so testing this application will involve many steps from start up times and GUI (Graphical User Interface) to measurement refreshing of rapidly changing data and visual representations of the data being transferred by the physical voltmeter. The application will first be tested based on the start up speed and consistency of the application opening across multiple devices. Although available hardware resources will impact opening and performance times, it is imperative that the application can open without crashing across multiple testing platforms to ensure widespread functionality and the lack of “bugs” in the software development of the application.

The GUI and menus are the next important activity to be tested as this is how the user interacts with the application and sets personal parameters to allow for a streamlined and personalized experience while using the application. The GUI will be tested by having data available to be changed by the user, such as units of measurements, etc. in order to make sure the application is responding to the user’s requests, as well as the links between buttons and launching new “activities”. These “activities” refer to new screens or paths being launched or created during a push of a button. If any of the buttons or lists are shown to be nonfunctional, then an activity listener has failed and likely a crash or memory dump will occur. This means all paths must be created, instantiated, and mapped correctly to ensure optimal functionality. Although startup and user interface are drastically important components of this application, the perhaps most important component is the data collection and representation from the digital voltmeter hardware to the application in the portable device.

The connection between the digital voltmeter and the portable device will be achieved using a Bluetooth® wireless connection. This Bluetooth® connection must first be granted permissions through the Android operating system in order to receive data. The connection of the Bluetooth® devices will be tested across multiple devices, i.e different mobile phones all using the application. Different versions of the Bluetooth® wireless technology will be tested in order to ensure backwards compatibility and the use of multiple mobile devices will present many different operations required for pairing the Bluetooth® transmitters. Not only will initial connection be an issue, but sending and receiving data is only present if both the Android operating system and user allow this communications path to be secured. Because the microcontroller will be used to compile and transfer data to the application, predetermined circuit voltages will be used to test the data transfer between the application and microcontroller.

The user will decide when data is transferred and represented on the application so every test can be controlled based on the values chosen to be test values. Finally the data received will be tested and compared for consistency and accurate graphical representation. The data must be transmitted and received first to ensure packages of information are able to be sent through the path between the microcontroller and the application. The most accurate way to test this data received is to compare the incoming numerical values measured to a preexisting voltmeter that has already been mass manufactured and calibrated. These results must compare accurately and consistently in order for the testing to prove a fully functioning prototype of the digital voltmeter. After this testing is confirmed, a final version of the voltmeter may be manufactured.

8 Administrative Content

8.1 Milestone Discussion

The following is a table which includes a list of initial milestones for designing the Smart Digital Voltmeter:

Task	**Deadline**
Senior Design I (Fall 2016)	
• Discussion of ideas and selecting a project idea. • Basic research on project components.	09/06/2016
• Submit initial project document - Divide and Conquer	09/09/2016
• ½ hour meeting with Dr. Lei Wei to discuss initial project document	09/19/2016
• Update Divide and Conquer files	09/30/2016
• Research hardware components • Choose microcontroller and other components needed	10/11/2016
• Order and test parts	10/20/2016
• Design prototype • Develop code	11/01/2016
• Submit Table of Contents	11/04/2016
• Current Draft of Senior Design I Documentation	11/11/2016
• ½ hour meeting with Dr. Lei Wei to discuss project draft	11/16/2016
• Final Document Due!	12/06/2016
Senior Design II (Spring 2017)	
• Build prototype	TBD
• Test and redesign project	TBD
• Finalize prototype	TBD
• Peer presentation	TBD
• Final report	TBD
• Final presentation!	TBD

Table 8.1: Project Milestones

For Senior Design I, our main goal is to finish the course with a working voltmeter assembled onto a breadboard. This voltmeter is then expected to send data wirelessly onto a smartphone via an Android application. We took small steps throughout the semester to accomplish this. Like any project, designing the smart digital voltmeter began with extensive research on hardware parts as well as software that would be used. The majority of our research consists of studying previous voltmeter designs and constraints. In addition, to make sure that our product can compete with the leading voltmeters in the market, we will research how to keep our digital voltmeter within 1% accuracy, low difference in cost, and little or no decrease in performance.

The main challenge that the group faced is the time constraint of designing a multimeter (voltage, current, resistance) in such a short amount of time. As a result, the project was simplified to designing a voltmeter capable of reading AC and DC voltages rather than a full three function multimeter. Voltage is, after all, the most important and easiest component to measure.

8.2 Budget and Finance Discussion

Component	**Quantity**	**Cost per Unit**
9V - Batteries	2	$0.49
Microcontroller	1	$10.99
PCB	2	TBD
Tablet / Phone	1	(already own Tablet / Phone)
Wireless Adaptor	1	(already included in Tablet / Phone)
Bluetooth® Module	1	$7.39
PTC Thermistor	1	$1.24
LCD	1	$1.99
Diodes	1-2	$0.34
Resistors	20	$0.08
Varistors	3	$1.10
High-Power Wirewound Resistor	1	$1.25
Trim Potentiometers	2-3	$0.35
Range Selector Switch	1	$0.47
Pushbuttons	2	$0.20
Power Switch	1	$0.13
LEDs	4	$0.10
Crystal Oscillator	1	$1.24
Capacitors	4	$0.10
P-channel MOSFET	1	$1.61
Estimated Total		**$35.24**

Table 8.2: Bill of Materials

The main specification under the budget and finance discussion is to keep the smart digital multimeter as affordable as possible. Therefore, the parts used will be the cheapest we can find on the market. The above hardware budget table includes multiple copies of parts and in essence includes more parts than required. As a result, the final estimated total is expected to decrease below the $59.39 estimated budget.

The above costs, as well as the components listed above, and even the quantities of each component are expected to change. The hardware budget table is a very rough estimate of what total budget we are expected to work with.

Software/License	**Cost**
Android SDK	FREE (open source)
Code Composer Studio	FREE (Texas Instruments)

Table 8.3: Software Budget

The microcontroller and the application will both be developed and designed respectively using free software such as the softwares listed above. These softwares will be used for the majority of the software development of this project, however if more software becomes necessary a free software will be used and listed above as well.

8.3 Sponsor Information

Voltmeters such as ours are used by a wide range of users such as electronic hobbyists, engineers, and electricians. Our group met with the owners of Commercial Lighting Enterprises Incorporated located in Orlando FL to discuss our project and their interest in possible sponsorship. This company has been operating for 22 years and serves clients such as Duke Energy and the Orlando International Airport. After pitching our idea, the owners LaChelle and William E. Wright happily agreed to fund our project with a budget of $500. The contact information for the owners and company are listed below.

Main number: 407-788-0075
Website: www.commerciallightingorlando.com
Address: 8130 N. Orange Blossom Trail Orlando, FL 32810

9 Product Operation

This section describes the full use of our final product. in Section 9.1, we will describe to install the smartphone application and outline the details for full usage. Section 9.2 will explain how to take a DC measurement and how to interpret the results on both the LCD and smartphone application. Similarly, in section 9.3 we will describe how to take an AC measurement and interpret the results. We will also discuss the features and requirements for safe operation in each section. Lastly, we will give some insight to troubleshooting in the case of incorrect measurements or product deterioration.

9.1 Using the Smartphone Application

My Smart Voltmeter is the accompanying application that will allow users to view and record voltage measurements with The Smart Digital Voltmeter. To begin using your product, visit the Google Play Store on your Bluetooth® capable Android device and download My Smart Voltmeter. Once installation has completed, pair your Bluetooth® capable device with the digital voltmeter hardware. Finally, you may open the Smart Digital Voltmeter application and select which voltage measurement you would like to measure, either AC or DC. Now you can measure the circuit as normal by using the two terminals on the voltmeter. These measurements will be displayed on the application based on the measurement the user selected. If AC voltage is selected, the user can choose between a numerical view or a graphical view.

9.2 Taking Voltage Measurements

9.2.1 Setup

Now that the smartphone application has been installed and a Bluetooth® connection has been established, the next step is to setup your device and take a voltage measurement. Begin by inserting a 9V battery (1) in the meter by connecting the positive and negative terminals to the connector clip (2) and move the power switch (3) from OFF to ON. Ensure that the green LED (4) is illuminated meaning that the device is turned on. The next step is to insert the red test lead (5) into the VIN jack of the meter, then insert the black test lead (6) into the COM jack. These will be used to take voltage measurements of the device under test (DUT).

9.2.2 DC Measurement

To take a DC voltage measurement, follow the steps below.

1) Ensure that the blue LED under “DC” is illuminated. This means that the device is in DC measurement mode and you are ready to begin taking a measurement.

    a) If the LED under “AC” is illuminated, press the “AC/DC Selector” button (7). This is used to switch between AC and DC modes.
    b) The blue LED under DC should now be illuminated
2) Select the range to measure by moving the appropriate slide (1, 2, 3, or 4) from the bottom to the top (ON) position on the “Range Selector” switch (8). Ensure that only one switch is up, as incorrect measurements will result from more than one range being selected.
3) Using the test leads, gently probe the DUT from one point to another while simultaneously looking at the LCD (8).
4) The LCD will display the real-time voltage (V), the average voltage (Vavg), the maximum (Vmax) voltage and minimum (Vmin) voltage
5) Remove the test leads from the DUT and view the smartphone application.
6) The application will display the values measured much like the LCD, presenting the real-time voltage (V), the average voltage (Vavg), the maximum (Vmax) voltage and minimum (Vmin) voltage.

9.2.3 AC Measurement

To take an AC measurement, follow the steps below.

1) Push the “AC/DC Selection” button and ensure that the blue LED under “AC” is illuminated.
2) repeat steps 2-3 of section 9.2.2 as these will be the same for both AC and DC modes.
3) The LCD will display the RMS voltage (V_{RMS}), the peak voltage (V_{PEAK}), and the frequency of the signal.
4) Remove the test leads from the DUT and view the smartphone application
5) The application will allow two different forms of data representation. The user and choose between numerical which presents all the data the DC measurement does such as real-time voltage (V), the average voltage (Vavg), the maximum (Vmax) voltage and minimum (Vmin) voltage. The user can also choose the graphical representation which will plot the values of the voltages measured, essentially acting as an oscilloscope.

9.3 Troubleshooting

If for any reason the device is not operating as defined in this section, please refer back to the above subsections and ensure all steps have been completed in the order in which they are listed. If the device is still malfunctioning or you believe a measurement to be incorrect, please consider the following options to help troubleshoot your device.

9.3.1 LEDs and Display

- Green power LED is not illuminating
    - Ensure that the battery has been inserted correctly
    - If still not illuminating, replace the battery

- Red LED under “Range Selection Warning” is illuminated
    - Return all pins of item 8 to the bottom positions
    - Select only one pin for the appropriate range
- No DC measurement values on LCD
    - Ensure DC mode has been selected with item 7 and blue LED is illuminated under “DC”
    - Ensure only one range has been selected and the red warning LED is not illuminated
    - If still no values are being displayed, this means that the voltage being measured is negative
    - Remove both leads and switch positions on the DUT, not the meter.
    - If in the correct measurable range, the LCD should now display the measurement value
- LCD displaying “Maximum Voltage!”
    - This means that the voltage being measured is above the selected range
    - Remove the leads from the DUT, select the next highest range and return the probes to the DUT
    - If still being displayed, repeat the previous step
    - If still being displayed when the 1000V range has been selected, the voltage being measured is above the maximum measurable level

9.3.2 Incorrect Measurements

If you believe that the meter is producing incorrect measurements, please first refer to the previous subsection and ensure all steps and user requirements have been completed. If this is the case, follow the steps below.

1) Use a proven voltmeter to measure the voltage.
2) If the value is off by a substantial amount, this means that either the safety features have been activated and halted measurement, or a component within the product has been compromised.
3) If this is the case, the product must be returned for repair
4) If the values are only slightly off, open the device packaging and locate the small blue potentiometer (10)
5) Re-probe the DUT after recording the correct value taken in step 1.
6) Using a small flathead screwdriver, very slightly adjust the potentiometer until the voltage measurement matches the value recorded in step 1.
7) Reassemble the device and resume normal use

9.3.3 Application Issues

If you are experiencing issues using the application, please follow the steps listed below.

1) Ensure that the correct application has been installed

2) Power on the meter and establish a Bluetooth® connection
    a) If this is issue, this means that the meter is defective and must be returned for repair.
3) DC mode not showing values
    a) Ensure the meter is also in DC mode by checking that the indication LED under “DC” is illuminated
    b) If issues are still present, the product must be returned for repair.
4) AC mode not showing values
    a) Repeat steps (a) and (b) above but the LED should be illuminated under “AC”

10 Project Summary

From the beginning of this project, we had an overwhelming amount of ideas and applications for our product. At the very first stages, we wanted to create a digital multimeter that could measure voltage, current, and resistance. We wanted to take away the hassle of reading and recording measurements by wirelessly transmitting the data to a smartphone application where the user could save the measurements and plot them against time and frequency to act as a sort of mini-portable oscilloscope. After researching all of the different existing designs, parts, and considering all of the constraints that would be against us, we decided to implement only voltage measurement capabilities. As digital multimeters have been around for many decades, the designs and technologies have been thoroughly tested and revised for optimal operation. These devices are very complicated and require a level of electrical, mechanical, and computer engineering that we have simply not reached yet. Additionally, we had to consider our largest constraint which was time, as we only had a few months to fully design our product. In the end, we determined to make a voltmeter that can measure both AC and DC voltages. To compensate for the lack of functionality compared to advanced multimeters, we knew that we could use our engineering skills to give the users more useful measurement information such as average, minimum, and maximum values that you would not normally see. And of course, because the smartphone application was the backbone of our project, we decided to continue with this feature and include all the bells and whistles to accommodate the meter.

Once we had established all what we wanted out product to do, we continued with our extensive research and parts selection. We came across several obstacles and flaws throughout the design process and made several changes as the project continued, all the way up until the final design. Even still, as we have not yet implemented the full design and tested all functionality, we expect to run into a few more roadblocks that will need to be addressed. Regardless, we are all confident with the final design and know that with the resources available and our own engineering skills that we can overcome any issues presented in the implementation process. The following list outlines all of the functions and features that our final product will offer:

1. AC and DC voltage measurement from 0-1000V
2. Measurement values displayed on an LCD
    a. Real-time voltage
    b. Maximum voltage
    c. Minimum voltage
    d. Average voltage
    e. RMS voltage of AC signals
    f. Peak voltage of AC signals
    g. Frequency of AC signals (to a certain limit)
3. Connectability with a smartphone application

   a. Read all aforementioned values
   b. Plot AC signals versus time and frequency
   c. Export data to be shared with associates
4. Guaranteed user and product safety through strategically selected hardware and test leads

In conclusion, throughout the process of this project, including the research, testing, and troubleshooting, our group learned a lot of valuable information. When it comes to the electrical hardware and circuitry, we had to apply all of the knowledge that we have learned throughout our years of electrical and computer engineering studies. Technologies such as varistors and thermistors used for input protection were topics that we had to research and master on our own as they were not covered in our classes. Existing knowledge of resistors, capacitors, diodes, transistors, op amps, and many others were put to the test throughout the design process. When it comes to the application and microcontroller, we had to use all of the various techniques learned to optimize our design. Though we were all tasked with the research and development of individual sections for the final product, it was a group effort that required the input and insight from each member. Throughout the stress and heartache, we acknowledged and overcame all of the obstacles in one way or another and came to a final design which fully covers all of the engineering knowledge we have acquired thus far. We are all fully confident that by the end of next semester, we will have a working product that will work as described above.

11 Senior Design II Summary

In the beginning of the Senior Design II semester, our team began preparations as planned. All parts had been ordered at this point and we were beginning to build the breadboard prototype when several obstacles presented themselves. In this section, we will outline the challenges faced and what was done to overcome them. Section 11.1 will describe the hardware design challenges, while section 11.2 will describe the microcontroller and application design challenges. Section 11.3 will conclude the paper by illustrating the final design, accomplishments, and unsolved issues.

11.1 Hardware Design Challenges

During the breadboard prototype construction, several obstacles were presented that would ultimately need addressing. The first issue arose with the ADC reference voltage. As shown in Fig. 6.6: Final Schematic, the voltage divider used 400kΩ and 100kΩ resistors to obtain the 1.0V voltage. Unfortunately, these values were not producing the 1.0V due to unknown reasons. We decided to go with 22kΩ and 6.8kΩ as the final values which gave us 1.014V at the input of the AREF pin.

The next issue was the voltage divider network in the input circuitry. Although the values listed in Table 3.2 are correct, getting those exact values proved quite difficult. We opted to implement a potentiometer in series with each of the “R2” dividers for precision tuning.

The following issues for the hardware design did not necessarily hinder the device operation, but after much consideration the team opted to remove some of the features that were initially desired. These included the MOSFET for the power supply protection and the indication LEDs. Because the LCD would warn the user when a measured voltage was higher than the range selected, the LEDs were not necessary and would only drain the battery faster. Additionally, the MOSFET that would protect the user from inverting the battery was not needed as the 9V battery clip we selected cannot be plugged in backwards.

The final remaining hardware changes came towards the end of testing. Once the standalone microcontroller prototype had been set up, we noticed that the ADC output would jump around a little bit when measuring a steady voltage source. As our input is comprised of only passive components, we knew that the problem must lie within the ADC operation. After further examination and research, we determined that the AVCC pin which is tied to VCC needed a low-pass filter to produce a steady DC supply to the analog comparators. We then added a single 100nF capacitor between AVCC and ground to filter the high frequency noise and our problem was solved.

Once the above changes had been made, it was time to order the PCB. Upon arrival and soldering, we immediately noticed an issue. The LCD was not displaying any text that we would write to it, regardless of the rest of the circuitry. We measured each of the pins of the LCD and discovered that the potentiometer which controls the backlight behind the text to make it visible was incorrectly routed in the schematic. This seemed to be our only problem so we implemented a white-wire modification to feed in the correct potentiometer operation which corrected the issue. Although we could have stayed with this modification, we noticed a few other aesthetic issues with the board such as switch labels. For this reason we redesigned the schematic and PCB and placed the order.

The last problem to be addressed was the Bluetooth setup. When testing the Bluetooth module functionality, we found that the microcontroller could not be programmed while the RX and TX pins were occupied. For this reason, we decided to use digital pins 9 and 10 of the microcontroller for the receive and transmit pins. In the MCU code, we assigned these pins to the variables and located a library online which takes care of the PWM transmission blocks. After implementing these changes into the code, we successfully established a connection that could receive and transmit data to a Bluetooth terminal on a smartphone. The final schematic for the project is shown below along with the final PCB design.

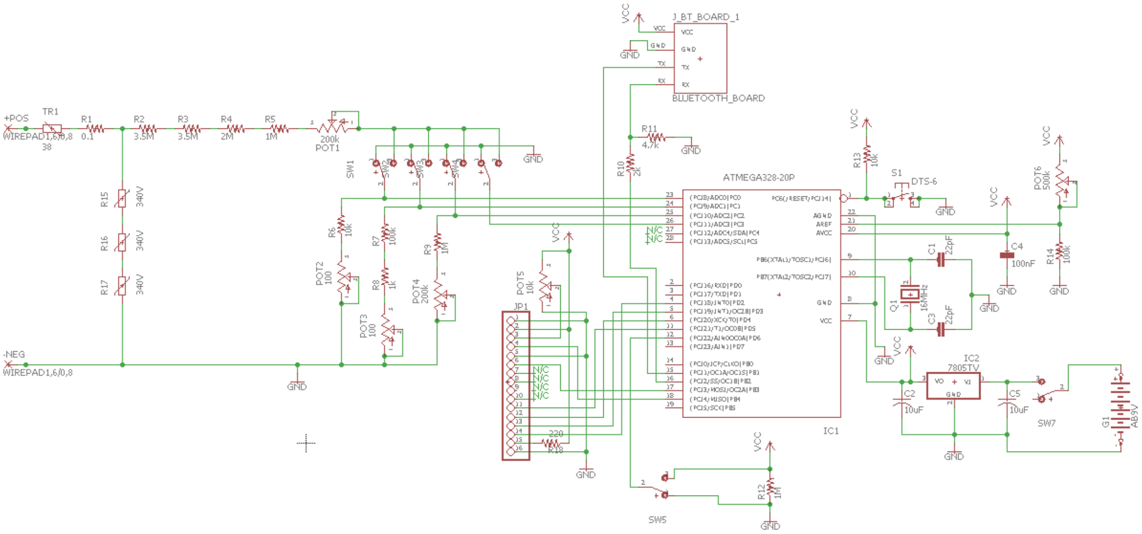


Figure 11.1: SD2 Final Schematic

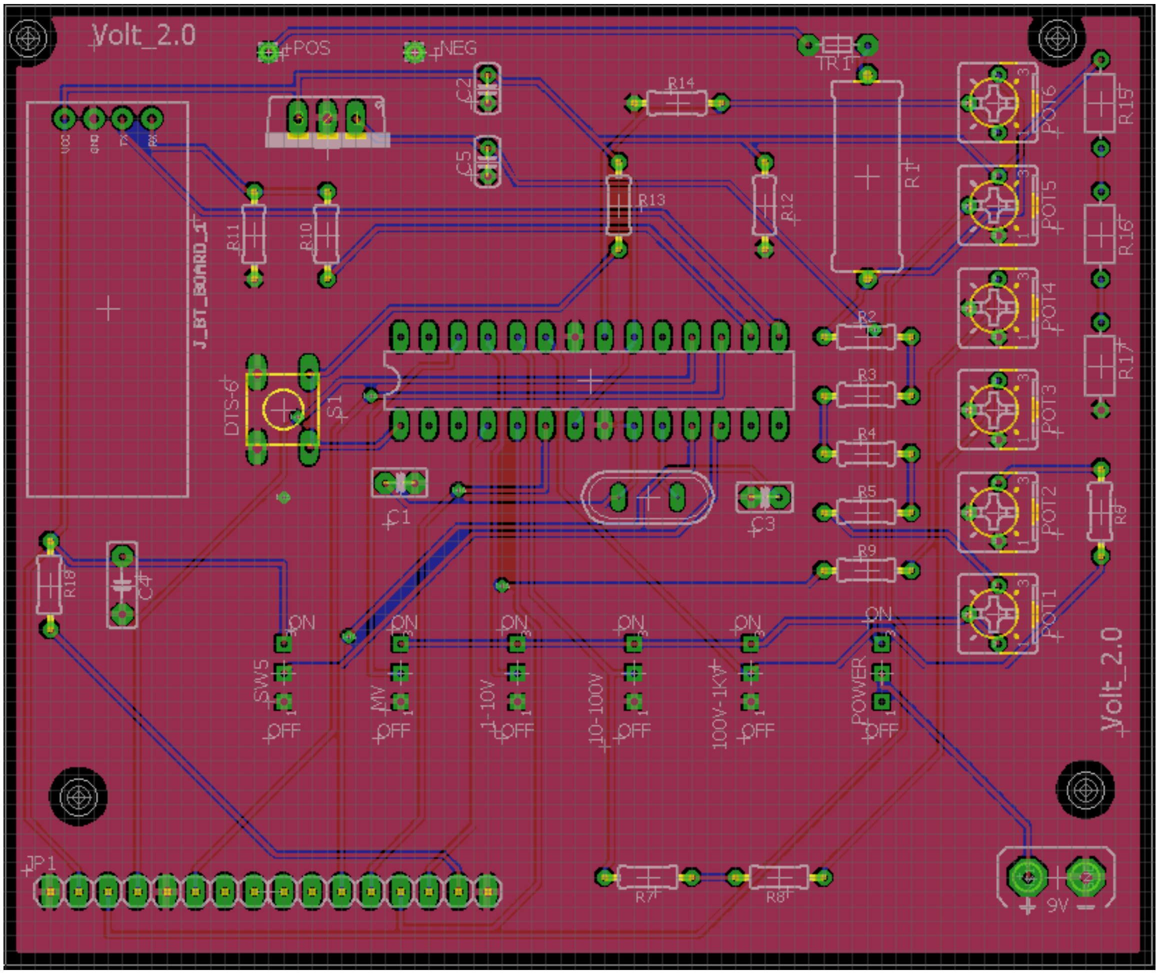
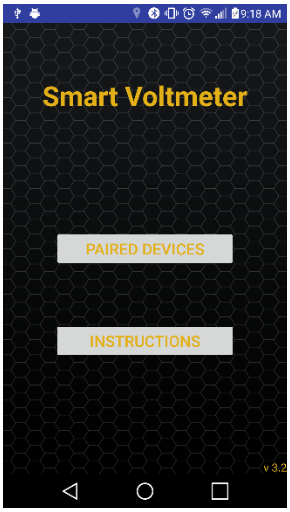


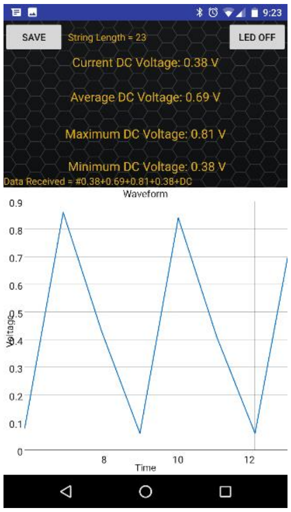
Figure 11.2: Final PCB Design

11.2 Application Design Challenges

Updated Application Issues and Design

Many iterations of the Android application were completed before deciding on the final design. The main issue in creating a fully functional Bluetooth enabled Android application is due to the life cycle of Android activity's life cycle. When an application's screen is changed, the information and commands on the previous screen are cleared to create available resources for the new activity. This means that connected Bluetooth devices would have the information discarded unless properly handled. This greatly impacted the design of the application and allowed a more streamlined and user friendly design. Major components of the original application were kept in the final design; however, the views were changed to allow the user to view both the numerical and graphical interfaces on the same screen. This keeps the Bluetooth data intact for the duration of the testing and data representation. The image below shows the updated design with the new data representation screen. The button to save the data to a text file is still implemented and functional as in the initial design.





String Length = 23

Current DC Voltage: 0.38 V

Average DC Voltage: 0.69 V

Maximum DC Voltage: 0.81 V

Minimum DC Voltage: 0.38 V

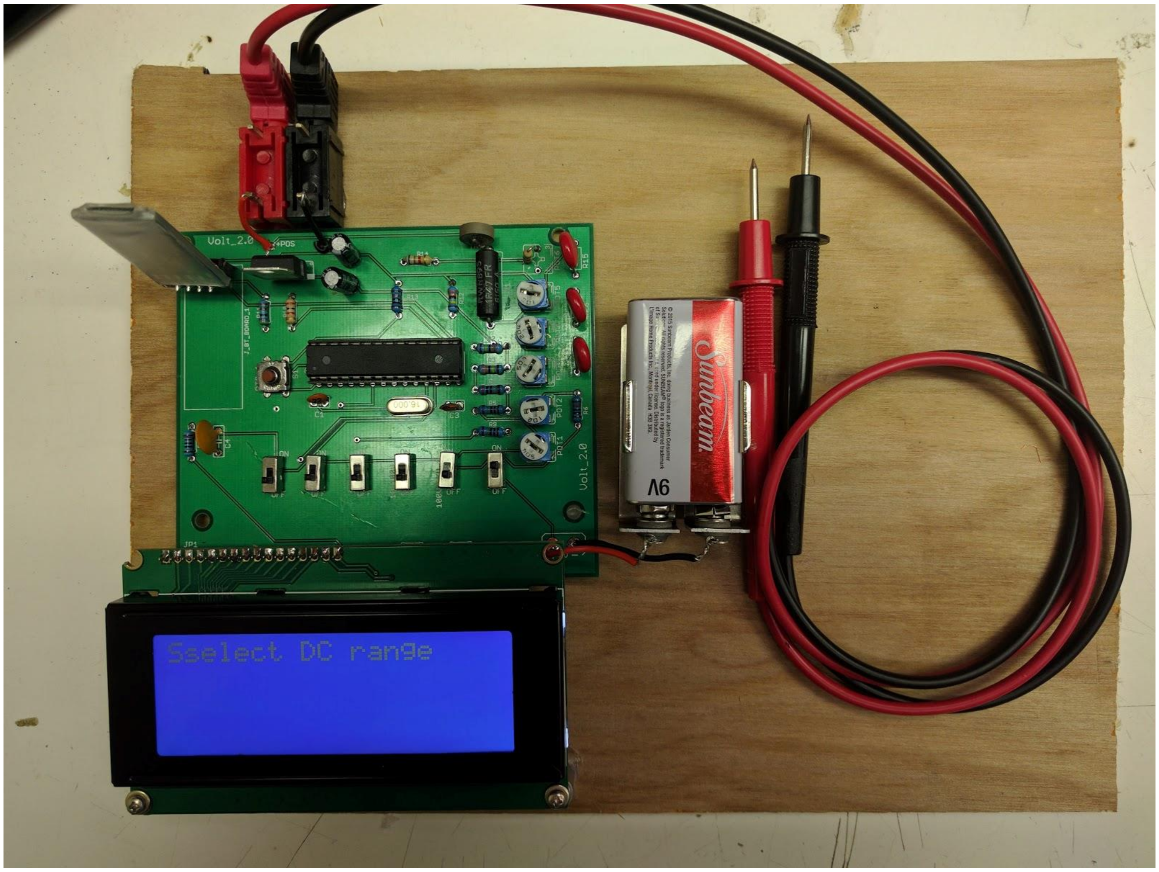
Data Received = #0.38+0.69+0.81+0.38+DC

11.3 Accomplishments and Unsolved Issues

With the completion of the Smart Digital Voltmeter, a list of key accomplishments as well as challenges can be observed. The team strived to construct the best functioning voltmeter given the allotted two semesters' time constraint. First, the main accomplishment of this project is a successful accurate measurement for all DC voltage ratings, including millivolts range, 1-10V range, 10-100V range, and 100-1000V range. The last range was assumed to function properly due to the first three voltage ranges performing flawlessly and being within 1% of our accuracy goal of other voltmeters in the market. A high enough voltage source was not found to completely test this fourth DC voltage range. Second, the Smart Digital Voltmeter successfully paired and transferred data via Bluetooth connectivity. Initially, difficulties with Bluetooth were encountered. The Bluetooth module could receive commands from the smartphone application but not transmit data to the application screen. This was a small user error with the circuitry leading from the transmit pin of the HC-06 Bluetooth module. This introduces the third accomplishment: Application Development. A successful connection with a custom-created Android Smartphone Application was obtained with the voltmeter. Data could then be easily sent from the voltmeter to any Android phone with this application installed. The fourth main accomplishment following this project is PCB design. Prior to Senior Design, none of the group members had any experience with PCB design. After multiple designs and two orders of PCB's from Elecrow's online PCB manufacturing store, a successful printed circuit board was constructed for the Smart Digital Voltmeter. Elecrow is a relatively inexpensive PCB manufacturer from China, with excellent customer service and speedy delivery. This high-quality, fast, and reliable PCB manufacturer played a great role in allowing the group to meet all deadlines. Finally, the group managed to expand on application design by providing a waveform display of AC voltage measurement. With this waveform, the team aimed to offer a portable voltmeter that could act as a pocket-sized oscilloscope.

Though the team accomplished most of what it set out to do, there were several challenges encountered. For instance, the Smart Digital Voltmeter's Application software through Bluetooth's wireless integration is only compatible with Android devices. To provide compatibility with iPhones and other apple devices, a license would be required through Apple for the permission to create Apple software applications. Second, there were several small issues when attempting to pair the voltmeter to earlier Android versions through Bluetooth via the custom Android application. Third, PCB safety isolation design slots were not performed, due to the expensive cost of implementing custom cuts onto the PCB. However, since the Smart Digital Voltmeter is targeted primarily for CAT II rating, or hobbyists/indoor use on smaller and safer circuits, these PCB isolation slots could be circumvented. The fourth challenge faced was the AC accurate readings being limited to a frequency of 100 Hz. This has to do with ADC sampling rate of the chip being a limiting factor. This leads to the next challenge of obtaining a smooth enough waveform display. The application frame skip is the major cause for this, as well as the Bluetooth transmission rate.

All in all, the project was a success. Below is a picture of the final product:



Appendix A: Resources

http://www.electronics-tutorials.ws/resistor/varistor.html

http://www.electronicrepairguide.com/how-to-test-a-fuse.html

http://content.fluke.com/promotions/promo-dmm/0518-dmm-campaign/dmm/fluke_dmm-chfr/files/safetyguidelines.pdf

http://www.autoshop101.com/forms/h4.pdf

http://www.teach-ict.com/gcse_new/communication/wi-fi_Bluetooth®/miniweb/pg3.htm

http://computer.howstuffworks.com/wireless-network3.htm

https://www.scientificamerican.com/article/how-does-wi-fi-work/

http://www.telegesis.com/about-us/zigbee-overview/

http://www.techradar.com/news/phone-and-communications/what-is-nfc-and-why-is-it-in-your-phone-948410

http://searchmobilecomputing.techtarget.com/definition/Near-Field-Communication

http://www.howtogeek.com/167783/htg-explains-the-difference-between-wep-wpa-and-wpa2-wireless-encryption-and-why-it-matters/

http://www.mouser.com/applications/rf-wireless-technology/

http://olimex.cl/website_MCI/static/documents/Datasheet_TM1637.pdf

http://www.engineersgarage.com/articles/wireless_communication?page=6

https://diyhacking.com/arduino-Bluetooth®-basics/

https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

http://www.allaboutcircuits.com/projects/make-a-digital-voltmeter-using-the-arduino/

http://www.electroschematics.com/9351/arduino-digital-voltmeter/

https://developer.android.com/design/index.html

http://www.electrical4u.com/working-principle-of-voltmeter-and-types-of-voltmeter/

http://www.techrepublic.com/article/apples-swift-programming-language-the-smart-persons-guide/

https://www.upwork.com/hiring/mobile/android-vs-ios-which-to-learn-first/