Talking Multimeter

Dustin Ha, Ibraheem Abdulazeez, Leonardo Torres Gonzalez, Affan Zafar
Electrical Engineering Department, Volgenau School of Engineering
George Mason University
Fairfax, VA, USA
Dha7@gmu.edu, Iabdulaz@gmu.edu, Ltorresg@gmu.edu, Azafar6@gmu.edu

***Abstract*—This paper presents the design and development of a fully accessible multimeter designed for visually impaired users, where the basis for the development involved communication with a visually impaired client. The device measures inductance, capacitance, and resistance, which is then output through audio feedback. Users will interact with the multimeter using buttons or voice commands, allowing seamless hands-free operations that give them an independence that they may not usually have. The system is portable, equipped with a rechargeable battery and built to withstand daily use on a workbench. Primary functions include a text-to-speech engine for real-time measurement outputs and a voice recognition system for mode selection. Attention to detail was given to create a simple, intuitive, and accessible user experience without relying on typical visual displays. The test demonstrates that the device accurately measures the standard electrical components in the chosen measurement operation. Future enhancement plans include expanding the range of commands and automating the unit functionality.**

I. Introduction

The basis for the project revolves around an electrical engineer hobbyist, who is legally blind and continues to work in the field of electronics. Conversations with the client made it clear that basic measurement tasks with resistance or capacitance can lack independence and typically rely on assistance. While some talking meters do exist[1-5], they are often limited in functionality, expensive, or not suited for hands-on electronics work. Traditional multimeter rely heavily on visual feedback, making them difficult for users with visual impairments. Important information like mode selection, measurement range, and results are typically displayed visually, with little or no alternative feedback options. As a result, blind or low-vision users face significant barriers to independently using these tools, highlighting the need for accessible designs like a talking multimeter.

As a result, the objective behind this project was to design a multimeter that could fully support independent work for visually impaired users. Electronics is a field that is apparent in precision and hands-on testing, but the majority of measurement tools assume the user has the ability to see a screen. There is a significant gap in the market for affordable-or even relevant-accessible devices that can perform the range of measurements a hobbyist or engineer would need. Largely, an affordable desktop talking multimeter costs around $100 USD [2-5], unfortunately, they lack the ability to measure capacitance and inductance, nor do they offer portable functionality. Most solutions focus only on traditional current, resistance, and voltage measurements leaving out inductance, capacitance, and frequency measurements. With the limits in mind, our goal is focused on closing the gap by creating a fully accessible multimeter that has the capability to provide measurements of inductance, capacitance, and resistance, which are communicated to users by a spoken audio output. Additionally, tactile switches are added for easy mode selection as well as the implementation of simple voice recognition which allows users to control the multimeter simply by using speech commands.

The multimeter's design promotes ease of usage, durability, and affordability. The system was developed around the Arduino Nano platform due to the high-speed processing and built-in DAC capabilities for audio output that the platform already provides[6]. Circuit simulation and prototyping for the multimeter were done using PSpice, the PCB design was completed utilizing KiCad. Together, these tools allowed the team to create a flexible modular system that has the capacity to evolve as new features are added. Across the board, this project delivers a practical tool that aids in the removal of barriers non-accessible electronics create for the visually impaired community.

II. APPROACH

A. Block Diagram

Fig. 1 shows the hierarchal basis for the the talking multimeter, the measuring circuits in color are shown on the left, and a general user reference is show. All processing is done by the Arduino Nano, our micro-controller, where it outputs the final reading via audio. The entire system is then powered by a 5 volt power supply.

B. Component Selection

1) Microcontrollers: Fig 2. shows our design matrix for the microcontroller, we selected the Arduino platform as it provides a lot of open source aid, utilizes a programming language that we are familiar with and had a smaller form factor. Raspberry Pi was and still is a close contender, however, due to it's form factor we could not implement. Discrete MCUs were considered but ultimately it was decided that a development board would be better suited for speed and efficiency.

2) LC Design: Fig 3 is where we decided between how we would incorporate an oscillator design, seeing as we had already found a reputable circuit principle[7,8] we decided to see if it was practical to make our own oscillator with discrete components. Ultimately it was that due to the availability and

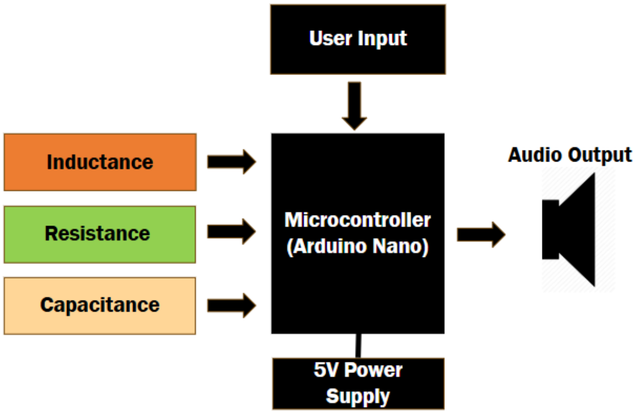

Fig. 1: Block Diagram

Microcontrollers	Ease of Programming	Low Cost	Familiarity	High Availability	Support	Form Factor	Total
Criteria Rating	4	1	4	2	5	4	
Arduino	4	5	3	4	5	4	
Weighted Rating	16	5	12	8	25	16	82
Rasberry Pi	4	3	3	3	5	3	
Weighted Rating	16	3	12	6	25	12	74
Discrete MCU	2	5	1	5	3	5	
Weighted Rating	8	5	4	10	15	20	62

Fig. 2: Microcontrollers Matrix

already wide range of oscillator op-amps on the market, that it was impractical to develop our own.

3) Audio Output System: Fig 4 shows our thought process behind either text to speech (TTS) or a pre-recorded audio solution. In this case it was decided a TTS solution was best as it didn't involve creating a separate library, the possibility of reaching memory limitations, and overall unnecessary complication that it would bring to the project. This all keeping in mind that a TTS solution includes the potential use of a large language model to supply the audio interface.

III. Measuring Circuits

The measuring circuits for resistance, capacitance, and inductance are separated into two discrete circuits. One for measuring the capacitance and inductance (LC Circuit) and the other for measuring resistance, these measuring circuits all output analog values that are then taken in by the Analog-to-Digital converter of the microcontroller to be processed. The processing steps of the microcontroller[6]-explained in a later section-is pivotal in achieving an accurate measurement.

Oscillator Circuit (OC)	Implementation	Low Cost	Familiarity	High Availability	Total
Criteria Rating	5	2	4	5	
Discrete IC	5	3	4	2	
Weighted Rating	25	6	16	10	57
Custom Built	3	3	4	2	
Weighted Rating	15	6	16	10	47

Fig. 3: Oscillator Matrix

Audio Output System	Voice Clarity	Customization	Ease of Development	Storage Requirement	Total
Criteria Rating	5	4	5	3	
Pre-Recorded Audio	5	2	3	3	
Weighted Rating	25	8	15	9	57
Text to Speech	4	5	4	5	
Weighted Rating	20	20	20	15	75

Fig. 4: Audio Output Matrix

A. LC Circuit

Fig 5 shows the basic design for the inductance/capacitance measuring circuit, everything coupled to the left of the C2 capacitor is the resonant tank circuit with the exception of C3, which is a low tolerance capacitor to be connected during calibration. Everything to the right is the oscillator portion of the circuitn, here comparator LM311[9] is biased so that it creates an oscillator, pin 2 operates at approximately 2.5V and initially operates at 0V, as a result the output at pin 7 is high[8]. Over time, pin 3 is charged via the C1 capacitor and reaches 2.5V where it causes a transience to occur at pin 7, this repeats over time causing a square wave to pulse continuously from the oscillator which can be seen in Fig 3 through the green pulse. At this point the tank resonant circuit responds to the pulse created by oscillator and can be seen in Fig 7 via the orange sinusoid. This principle is repeated throughout the operation of the LC circuit, and the subsequent operations are done largely through the microcontroller. In this initial stage the frequency is taken by the Arduino as:

$$F_1 = \frac{1}{2\pi\sqrt{L_r C_r}} \tag{1}$$

Where Lr and Cr are the resonant inductor and capacitors respectively. To find the operating values of Lr and Cr, the

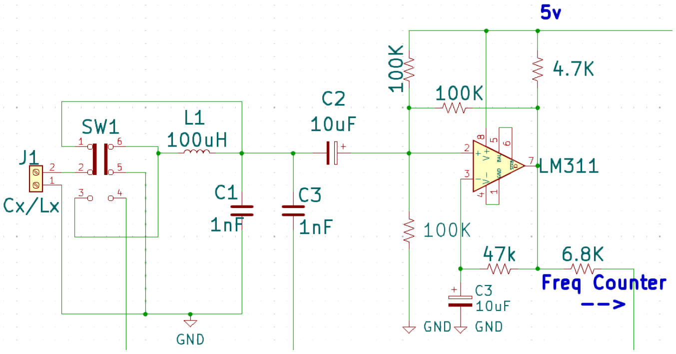

Fig. 5: LC Circuit

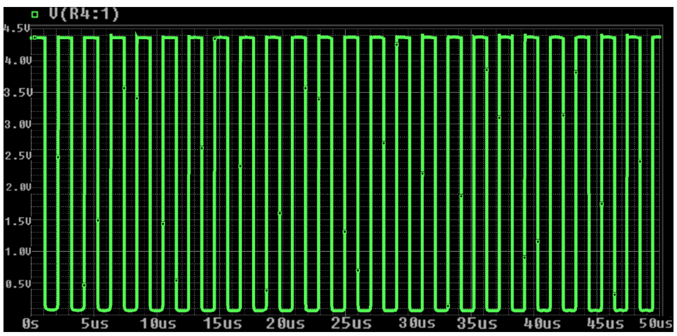

Fig. 6: LM311 Pin 7 Oscillation

calibration capacitor is connected, the frequency changes as a result:

$$F_2 = \frac{1}{2\pi\sqrt{L_r(C_r + 1nF)}} \tag{2}$$

At this stage the microcontroller solves for the operational values:

$$C_r = [\frac{(F_2)^2}{(F_2 \cdot F_1)^2}] \cdot 1nF \tag{3}$$

$$L_r = [\frac{1}{4\pi^2 \cdot F_1^2 \cdot C_r}] \tag{4}$$

Depending on the mode selected by the user, the capacitance and inductance can be measured which again causes the frequency to change and be stored onto the Arduino as:

$$F_3 = \frac{1}{2\pi\sqrt{L_r(C_r + C_x)}} \tag{5}$$

or

$$F_3 = \frac{1}{2\pi\sqrt{(L_r + L_x)C_r}} \tag{6}$$

Where Cx and Lx are the unknown capacitance and inductance values respectively, the Arduino then calculates the unknown values via:

$$C_x = [\frac{(F_1)^2}{(F_3)^2}] \cdot C_r \tag{7}$$

or

$$L_x = [\frac{(F_1)^2}{(F_3)^2}] \cdot L_r \tag{8}$$

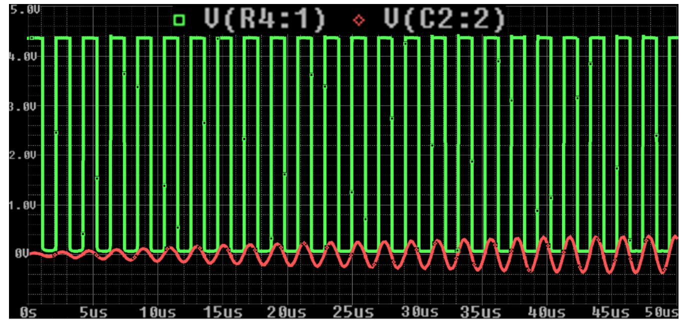

Fig. 7: Resonant LC Response

B. Resistance Measuring Circuit

The resistance portion of the circuit consists mainly of a BJT and back to back diodes, that produce a constant and stable current. The circuit is shown in Fig 8. The diodes are

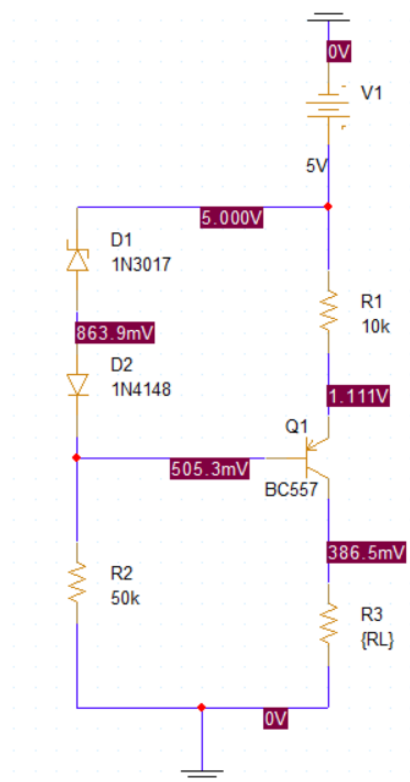

Fig. 8: Resistance measuring circuit, R3 is the resistor to be measured

implemented to reduce transience when the unknown resistor is connected. For Arduino processing we use a low tolerance resistor for R1, that allows us to declare it as a constant within the Arduino code, as a result we can process the voltage at resistance R1 to find the operating current when the unknown resistor is connected. The voltage is then read from the unknown resistor to find the unknown resistance value via basic Ohm's Law.

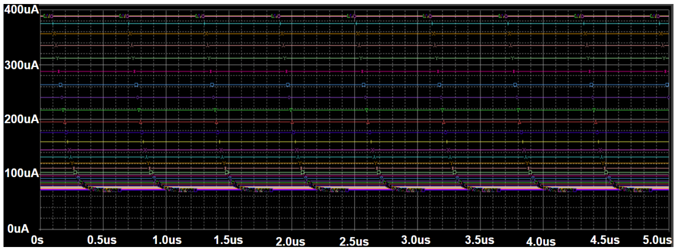

Fig. 9: Current values from varying resistance ranging from 1 Kilo-Ohm to 1 Mega-Ohm

C. Error

The LC measuring circuit in the ideal format has an error of ±0.059 femtofarad as previous testing has been shown by others who employed similar circuit structures[7]. As for resistance the error propagates due to the inconsistency in the R1 resistance value, connecting a low tolerance and temperature resistant resistor can solve this along with possibly editing the design to switch resistors depending on the requested range or intermittent calibration after a set period of time. Due to limited testing, an exact error propagation is not available yet.

IV. Power Consumption

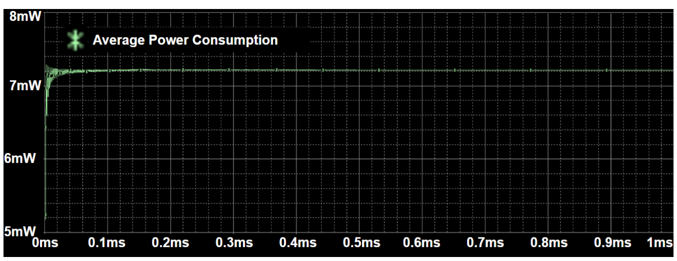

Fig. 10: Average power consumption of LC circuit

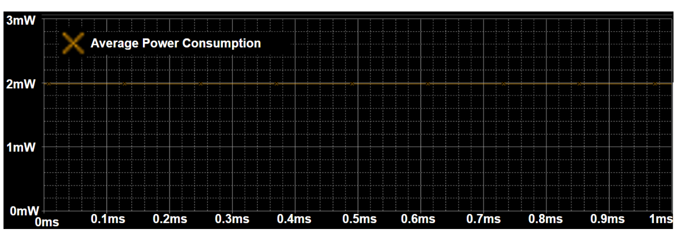

Fig. 11: Average power consumption of resistance circuit

Power consumption for the circuits is show in Fig 10 & Fig 11 where the consumption of the LC circuit is $\approx 7.2mW$ and the resistance circuit $\approx 2.0mW$, however, these are taken at their baseline and will fluctuate depending on the components tested. Regardless, total wattage is expected to be $\approx 176.2mW$ when including the Arduino[6]. When accounting for fluctuations caused by component testing, they can all draw power that the Arduino is comfortably capable of feeding.

V. Design Alternatives

There are many other designs available, however, due to the small range of values the client wishes to measure, specifically for inductance and capacitance, a lot of the available options are not ideal. Regardless, we'll explore some alternatives we considered when developing the circuits.

A. LC Alternatives

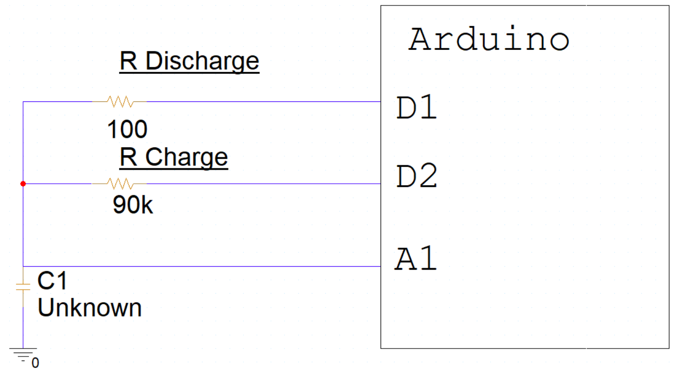

Fig. 12: RC Charge Circuit

For this configuration, D1 initially discharges the unknown capacitor to 0V. Afterwards, D2 becomes active and outputs 5V to the "R Charge" resistor and unknown capacitor, the large resistance reduces the charge rate of the capacitor allowing the data resolution to be higher. Once the voltage reaches 2.5V the Arduino switches to D1-set to ground-where the unknown capacitor is then discharged[10] via the "R Discharge" resistor. Throughout this entire process, the A1 pin is collecting data on the voltage with respect to time, thus allowing for the following equations:

$$e^{-\frac{\tau}{RC}} = \frac{V_S - V_C(t)}{V_S} \tag{9}$$

$$-t = RC \ln\left(\frac{V_S - 0.5V_S}{V_S}\right) \tag{10}$$

$$C_x = \frac{\Delta t}{0.6931471 \cdot R} \tag{11}$$

Where Vs is the supplied voltage, Vc(t) the capacitance voltage, and R the resistance value in the RC circuit. This approach is simple and a valid method for most capacitors, capable of measuring down to $\approx$ 5pF [10]. However, it loses out to the original method displayed out in this paper. The circuit lacks the ability to accurately measure inductance. Modifying this circuit to measure inductance could be done on the Arduino by instead feeding the inductor with a PWM signal via D1 or D2, and making A1 a frequency counter. The inductance value is then found via:

$$L = \frac{R}{2\pi F} \tag{12}$$

As a result, the approach is more compact and requires less components than the LC circuit. Unfortunately, questions of

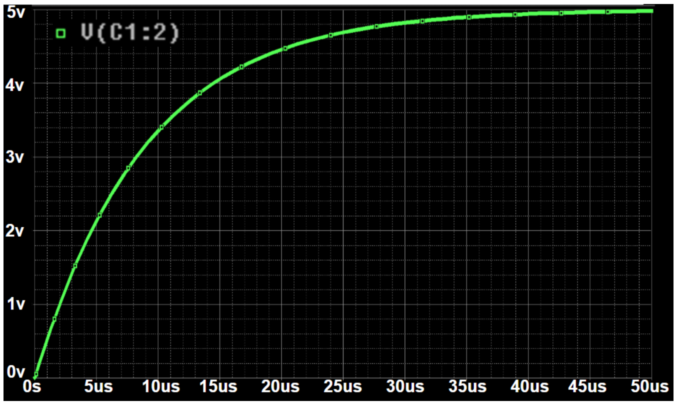

Fig. 13: Charge process of capacitor via R Charge resistor

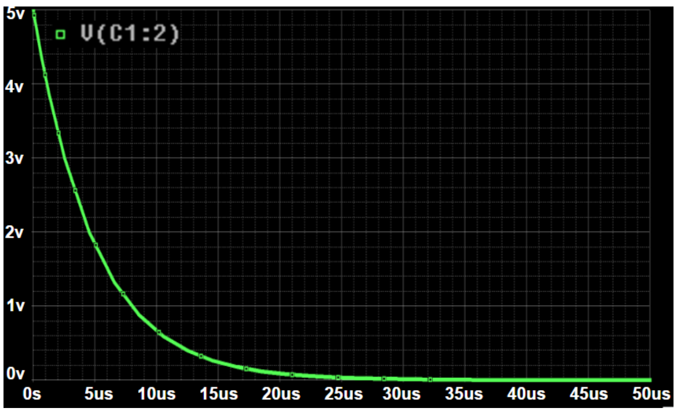

Fig. 14: Discharge via R Discharge resistor

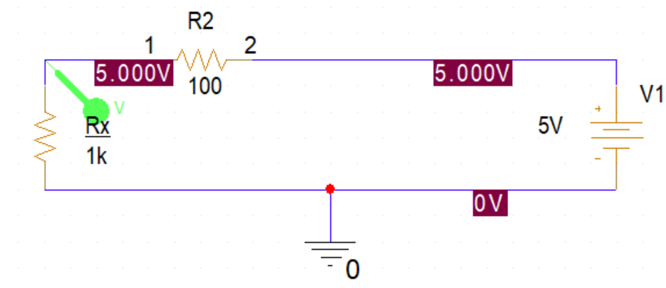

Fig. 15: Voltage Divider

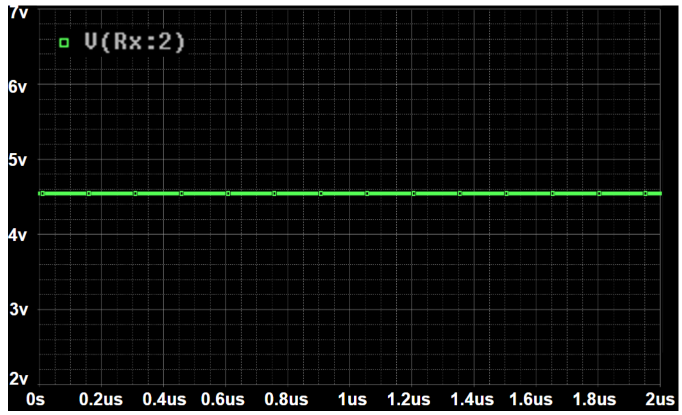

Fig. 16: Voltage measured for unknown resistor

accuracy come into play, as the error is dependent on the absolute accuracy of the components used in the circuit, something the LM311 circuit already solved. Additionally, introducing a discrete oscillator would aid in the accuracy of the inductor measurement, as they generate a clearer waveform. Then, since capacitors respond to waveforms similarly to inductors, the same exact circuit can be utilized to measure both. Taking this into consideration, it leads to developing something similar to the LC Circuit already shown, which trumps the notion of having a discrete circuit for each measurement type, as currently there is no other way to measure inductance besides utilizing a waveform.

B. Resistance Alternatives

Resistance measurements focus largely on it's DC properties, as a result, another method measuring resistance that was explored was a voltage divider. In this case the Arduino simply reads the voltage between the known and unknown resistor and results in the following equation for the unknown resistance:

$$R_x = \frac{\frac{V_o}{V_{in}} R}{1 - \frac{V_o}{V_{in}}} \tag{13}$$

This method is accurate when utilizing a microcontroller and is valid, however, it begins to falter as the resistance values increase since the Arduino is only capable of outputting 5V. The larger resistances begin to take a larger portion of the voltage which eventually overpowers the original resistance. A solution to this would be to have an array of resistors that can switched in when larger resistances are being measured, however, this comes at the cost of needing a significant amount of space. The former constant current setup still allows for a larger range of test values while keeping a low profile and can be biased to change the output current. Ultimately, it is for those reasons that the constant current circuit won out over the voltage divider.

VI. Audio Output System

A. Talkie Library Overview

For audio we're leveraging the Talkie library [11]. This library does speech synthesis using Linear Predictive Coding (LPC)[12], which happens to work well with the Arduino Nano. It comes with a built-in vocabulary of over 1000 words, including numbers and some technical terms[11]. The Talkie library doesn't produce a true analog audio signal directly. Instead, it generates the audio waveform using Pulse Width Modulation (PWM) and since the Arduino Nano doesn't have a built-in DAC, we create a simple one using a passive low-pass filter. The filter smooths out the high-frequency PWM pulses, leaving an approximation of the desired analog audio waveform [13]. Unfortunately, the signal coming out of this simple filter does not have enough power to drive a speaker directly so an audio amplifier must be implemented, making the signal strong enough to drive a small speaker, which then produces the audible sound.

B. Converting Numerical Values to Speech

The Talkie library works with predefined words[11]. To say a number, we need to break it down into its components (digits, decimal point, units, prefixes) and then tell the Talkie library to say each corresponding word from its vocabulary in the correct sequence. A specific algorithm within the software in the Arduino easily takes care of this.

VII. User Interface

A. Buttons

Per the client's request, the multimeter will have tactile buttons, a mode select, a calibrator/zero, and measuring range with LCR, CAL/0, and RNG in braille respectively to label the buttons. A switch will be implemented to power the device on or off with the words labeled braille on their respective sides. In addition, the device will also announce every action done by the user, again using the Talkie Library. The calibration/zero button works to activate the calibrator for the LC circuit or to zero out the resistance value measured. Again, any process is announced if actuated or done automatically by the Arduino. In the case for the mode select, when pressed once it announces the current mode and the subsequent two presses prompt the user whether they want to switch to the other modes, with a fourth press cycling back to the beginning. To select the mode the user would simply hold the button until the Arduino announces it has changed modes. The range button simply toggles the unit range and by holding a press it will go into automatic mode, selecting the best unit to speak aloud.

VIII. Software

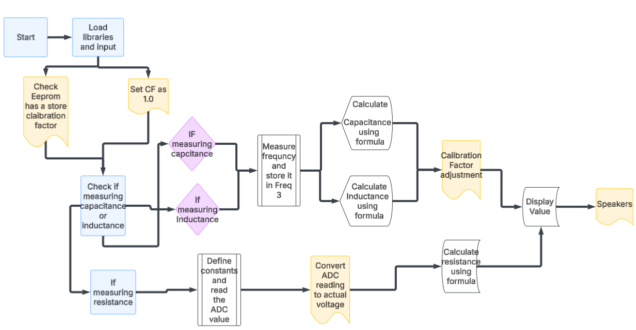

Fig. 17: Software Flowchart

The software flowchart lays out the specifics of our Arduino processing, for our case the software is the smaller portion for the project, at least in its current form. Essentially the Arduino will first decide between the LC circuit or resistance circuit and begin their respective processes. For the LC circuit it will check for the measuring mode of capacitance or inductance and begin the measuring frequency processes that were laid out in the "LC Circuit" portion of the report. For the resistance circuit, it will simply follow the process already stated in the "Resistance Measuring Circuit" portion of the report.

IX. Summary

The project as it is, has been successful in producing a functional prototype of an accessible multimeter, designed to aid users with visual impairments. The meter has displayed the ability to measure inductance, capacitance, and resistance. Additionally, device provides all results to users through spoken audio rather than the traditional visual displays people commonly use. Users have the ability to press a tactile switch with the multimeter, this offers an accessible and flexible experience for users with visual impairments. As a result, it addresses an issue for engineers who are visually impaired and offers them a more practical and inclusive tool.

X. Future Work

While the measurements and accessibility features on the multimeter have been implemented, there are still improvements that are needed in order to enhance the capabilities of the device. The intended future enhancements include the addition of Bluetooth or a USB interface, this allows for both firmware updates and PC-based data logging. A rechargeable battery is also planned to improve portability and eliminate the need for frequent battery replacements. There will be an expansion on the command library for voice recognition in order to support a user's need for flexible and detailed control. Auto-ranging capabilities are also in consideration as they have the ability to make measurements easier, along with adding a compact 3D-printed enclosure in order to provide extra protection for the device during everyday usage. Both self-check features and a spoken calibration mode are in planning; these will aid users in being able to independently maintain measurement accuracy without the help of an outside source. Looking ahead to the summer and fall, the team plans to enhance frequency detection in order to improve its accuracy across a larger range of signals. Further work will focus on incorporating the voice activation system so it has the capacity to handle commands more reliably within loud environments. The goal is the finalization of the printed circuit board allowing for the implementation of the 3D printed shell to securely place the entirety of the system, along with making the design easier to interact with. These planned improvements have the capacity to move the current prototype to a finished field-ready tool.

A. Potential Large Language Model Implementation

As of now, the team is considering implementing a large language model (LLM), TinyLlama, via a Raspberry Pi 5 that could significantly aid in the client's experience with the talking multimeter. This would allow for an audio assistant to prompt and be prompted by any user, essentially allowing for the client to have the equivalent of Amazon's Alexa controlling the multimeter functionality. As for details, current Raspberry Pi 5 (8 GB) gives us a quad-core 2.4 GHz Cortex-A76 [14] and still runs happily from the 5 V rail already available inside the project enclosure, leaving roughly 6 GB free for model weights once Linux is up and services are loaded. When we install the open-source speech stack— whisper.cpp for speech-to-text, llama.cpp with a quantised TinyLlama-1.1 B or Llama-3-8 B for language understanding, and Piper for text-to-speech—the whole voice loop stays offline yet remains responsive. Community tests show the Whisper "tiny-en" model can transcribe in real time on a single Pi core [14]; TinyLlama delivers around 4–5 tokens s on an 8 GB Pi 5, fast enough for command sentences, while even the larger Llama-3-8 B model runs at 2–3 tokens if we need richer language skills. Piper, optimized for the Pi, adds only a few-hundred-millisecond overhead and remains one of the lowest-latency TTS engines in edge benchmarks, so round-trip interaction typically stays under 1.5 seconds. To fit everything in memory and keep power draw low we store the LLMs in 4-bit GGUF format [14]; this single step slashes model size and RAM usage by roughly three-quarters with minimal accuracy loss, making it practical to host up to a 7- or 8-billion-parameter assistant entirely on the Pi's LPDDR4 while still leaving headroom for audio buffers and I/O tasks. Because every stage—from microphone capture to language reasoning to speech synthesis—runs locally, no user audio ever leaves the device, the system works where Wi-Fi is unavailable, and firmware or model updates can be side-loaded over the existing USB-C maintenance port [14]. In short, a Raspberry Pi 5 coupled with a quantized edge-LLM lets us replace the current fixed-phrase interface with natural voice commands without compromising cost, portability, or privacy.

REFERENCES

[1] D. V. Gadre, " Talking Multimeter and LCR Meter: Accessible for Blind or Visually Impaired Persons(VIPs)." Noida, Mar. 4, 2021

[2] J. Lo and R. Dipirro, "Talking multimeter," Talking Multimeter, https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2016/rdd49_hl887/TalkingMultimeter_rdd49_hl887/TalkingMultimeter_rdd49_hl887/index.html (accessed May 2, 2025).

[3] MPJA, "Talking digital multimeter," Electronic Components and Accessories, https://www.mpja.com/Talking-Digital-Multimeter/productinfo/35042%20ME/ (accessed May 2, 2025).

[4] Omega, "Hands free, high performance talking digital multimeter/thermometer," Omega Engineering, https://au.omega.com/pptst/HHM2.html (accessed May 2, 2025).

[5] Amazon, "Zoyi 5566se Digital Multimeter with Bluetooth speaker, AC/DC Voltage Tester, resistance, capacitance, temperature, frequency measurement, and voice broadcast for home and industrial use: Amazon.com: Tools & Home Improvement," Amazon, https://www.amazon.com/ZOTEK-Multifunctional-Multimeter-multimeter-integrates/dp/B0B2181BTQ

[6] Arduino, "Nano," Arduino.cc, https://docs.arduino.cc/hardware/nano/ (accessed May 2, 2025).

[7] A. Andijic and A. Tschurtz, "Development of an Electric Circuit to Evaluate Small Capacitance Changes," schiessle.de, https://www.schiessle.de/emt1/MessKleinKap/MessKleinKap1.htm (accessed May 2, 2025).

[8] N. Heckt, "LC meter project and kit - L/C Meter IIB," lc meter project kit, https://pe2bz.philpem.me.uk/Comm01/-%20TestEquip/-%20Tester-NonActiveDevice/Cl-251-IndCap-MeterKit/lc-meter-project.htm (accessed May 2, 2025).

[9] Texas Instruments, "LM111, LM211, LM311 Differential Comparators" LM311 Datasheet, Sept. 1973 [Revised March 2017].

[10] Djatmiko, Wisnu. (2019). Capacitance Measurements System Using RC Circuit. KnE Social Sciences. 3. 603. 10.18502/kss.v3i12.4131.

[11] A. Joachimsmeyer and P. Knight, "Talkie," Arduino, https://docs.arduino.cc/libraries/talkie/ (accessed May 2, 2025)

[12] H.-S. Kim, "Linear Predictive Coding is All-Pole Resonance Modeling," Linear predictive coding is all-Pole Resonance Modeling, https://ccrma.stanford.edu/h̃skim08/lpc/ (accessed May 2, 2025).

[13] K. A, "Arduino PWM Output Filter Circuit," Simple DIY Electronic Music Projects, https://diyelectromusic.com/2021/07/11/arduino-pwm-output-filter-circuit/ (accessed May 2, 2025).

[14] G. E. Dahl, D. Yu, L. Deng and A. Acero, "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30-42, Jan. 2012, doi: 10.1109/TASL.2011.2134090.