## Introduction:

In this lab, my objective was to design and implement an ohm meter using a microcontroller with a high degree of accuracy and independence from external devices like PCs or laptops. To accomplish this, we decided to create a battery-operated device or use a 5V wall power supply that could auto-range over a resistance range of 1 ohm to 1 mega-ohm. The desired resolution was 4-1/2 digits (0000 - 1999), and we aimed to achieve at least 1% accuracy. The primary components used in the design were an Arduino microcontroller, a LiquidCrystal I2C display,  a MUX switch, and an external 16-bit ADC from ADAfruit.

## Hardware Design:

The core of our hardware design was based on the principle of a voltage divider circuit. The voltage divider equation Vo = Vs(R2 / (R1 + R2)) relates the output voltage (Vo) to the supply voltage (Vs) and the known (R1) and unknown (R2) resistances. By varying R2 using a multiplexer (MUX), we could measure the voltage across each branch of the MUX and determine the most accurate resistance value.

To better illustrate the voltage divider circuit and the MUX operation, please refer to Image 1 and 2 below:

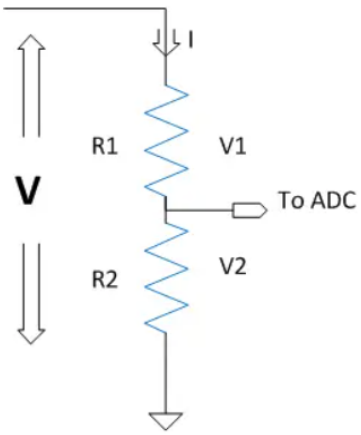



Figure 1:  Image of a voltage divider

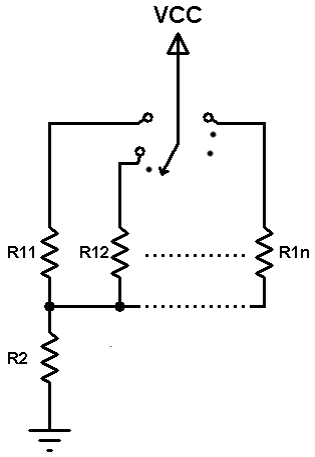



Figure 2:  Image of a MUX voltage divider

We carefully designed the schematic and PCB layout to ensure all connections were accurate and accounted for. The microcontroller was responsible for controlling the MUX, reading the voltage across different branches, and calculating the resistance using the voltage divider equation.

## Code Implementation:

The Arduino code played a critical role in the project's success. It was programmed to interface with the external ADC and control the MUX to select different branches. Here's an overview of how the code worked:

The code started by initializing the ADC and setting up the digital pins for the MUX control.
It utilized case statements to activate different branches of the MUX one by one.
For each branch, the code would read the voltage across it using the ADC and then calculate the resistance using the voltage divider equation.
The measured resistance values were then displayed on the connected LCD screen for easy observation.
Test Results:

During the testing phase, we connected various resistors with known values to the ohm meter and compared the measured values with reference measurements taken using a Keythley measurement system. The test results showed that our design achieved excellent accuracy, with the maximum error observed being around 2%. The worst-case error was well within the acceptable range of 20%.

We attribute the accuracy of our results to several factors:

The high-resolution 16-bit external ADC: By using an external ADC with superior resolution compared to the built-in ADC in the Arduino, we could obtain more accurate voltage measurements.

MUX and voltage divider design: The use of the MUX allowed us to select the most appropriate branch for measurement, ensuring that the resistors used were close to each other in value, minimizing error. The voltage divider circuit itself provided a stable and reliable method for measuring unknown resistances.

Careful component selection: We made an effort to choose resistors with low tolerance (1% error) to minimize deviations from the expected values.

Proper testing and calibration: We conducted rigorous testing and calibration to ensure the system was functioning optimally, and any potential sources of error were identified and addressed.

## Considerations in ADC Selection:

The Arduino microcontroller, which we used as the core of our ohm meter design, comes equipped with a built-in ADC. The ADC's resolution is 10 bits, allowing it to produce 2^10 or 1,024 discrete voltage values within its input voltage range. While the built-in ADC might be sufficient for certain applications, we needed to critically assess whether it met the stringent accuracy and resolution demands of our ohm meter project. Some considerations for choosing an external ADC would be the following.

Resolution: One of the primary factors in our decision-making process was the resolution of the ADC. A higher-resolution ADC can detect smaller voltage changes, allowing for more precise voltage measurements. With a 10-bit ADC, the voltage step size is limited, which can lead to inaccuracies when calculating resistance values for the ohm meter.

Accuracy: We also needed to consider the absolute accuracy of the ADC. Various factors can affect the accuracy of an ADC, such as the reference voltage, noise, and calibration. With a 10-bit ADC, the absolute accuracy might not meet the high standards required for our project.

Measurement Range: The limited resolution of the built-in ADC also restricted the effective measurement range. For an ohm meter, where we needed to cover a resistance range from 1 ohm to 1 mega-ohm, the built-in ADC's limited voltage steps might not provide the desired granularity for precise resistance measurements.

Sensitivity to Noise: A lower-resolution ADC might be more sensitive to noise, which could introduce errors in the measurements and compromise the overall accuracy of the ohm meter.

Selection of an External 16-bit ADC:

After careful evaluation and consultation with our peers and professors, we decided to use an external 16-bit ADC from ADAfruit for our ohm meter design. The reasons behind this decision are as follows:

Higher Resolution: The 16-bit ADC offers a much higher resolution than the built-in 10-bit ADC. With 2^16 or 65,536 voltage values within the input range, the ADC can detect smaller voltage changes, leading to more accurate voltage measurements.

Improved Accuracy: An external 16-bit ADC provides better absolute accuracy, enabling more precise resistance calculations using the voltage divider circuit. This was crucial in achieving our goal of 1% accuracy over the 1 ohm to 1 mega-ohm resistance range.

Expanded Measurement Range: The higher resolution of the external ADC allowed us to expand the measurement range without compromising on accuracy. This flexibility was essential for accommodating a wide range of resistances in the ohm meter.

Noise Tolerance: The external ADC was more robust against noise interference, contributing to more reliable and accurate measurements.

We opted to use an external 16-bit ADC instead of the built-in ADC in the Arduino for our ohm meter design due to its higher resolution, improved accuracy, and noise tolerance. By making this decision, we ensured that our ohm meter provided precise and reliable resistance measurements across the desired range. This choice allowed us to achieve the project's objectives and deliver a high-quality ohm meter with exceptional accuracy and independence from external devices.

## Accuracy Considerations in ADC Selection:

When choosing an analog-to-digital converter (ADC) for an ohm meter, it's essential to understand the difference between resolution and accuracy. Resolution refers to the number of distinguishable steps or

digital values that an ADC can produce over its input voltage range. On the other hand, accuracy is a measure of how closely the ADC's output matches the actual input voltage.

Consider a 16-bit ADC with an input voltage range of 0 to 2.5 volts. It offers a resolution of 2^16 or 65,536 discrete voltage values within the range. This gives a resolution of approximately 38 microvolts (38 μV) per step, which is the difference between two adjacent voltage values. However, while the ADC provides 65,536 voltage values, it only has 65,535 voltage increments or "steps" across the range.

The least-significant bit (LSB) of the ADC, the rightmost bit, has the smallest weight and represents the smallest voltage change that the ADC can measure. In the case of the 16-bit ADC with a range of 0 to 2.5 volts, the LSB has a voltage weight of approximately 38 μV. On the other hand, the most-significant bit (MSB), the leftmost bit, has the largest weight, representing a voltage change of approximately 1.245 volts, which is approximately half of the full-scale range (2.5 volts) of the ADC.

However, it's crucial to note that the resolution provided by the ADC's bit depth doesn't necessarily equate to absolute accuracy down to the LSB level. Many factors influence the ADC's accuracy, including electrical noise, the precision of the ADC's voltage reference, component tolerances, and other sources of error.

In practice, achieving 16-bit accuracy in ADCs is challenging due to various limitations, such as gain errors, reference voltage accuracy, and temperature dependencies. Even with a perfect voltage reference, the initial absolute accuracy of a 16-bit ADC may be closer to 14 bits. Therefore, it's more reasonable to assume an accuracy of 13 bits when choosing a 16-bit ADC.

With an assumed accuracy of 13 bits, the effective resolution of the ADC becomes 2^13 or 8,192 discrete voltage values within the input voltage range. This results in a resolution of approximately 0.122 millivolts (0.122 mV) per step. While this is still a very high level of precision, it's essential to recognize that the three least-significant bits might not provide accurate enough information for certain applications.

To improve the accuracy of the measurements, several strategies can be employed:

Use a High-Precision External Reference Voltage: External reference voltage sources with higher accuracy can be used instead of the built-in reference of the ADC to enhance measurement accuracy.

Component Selection and Calibration: Choosing high-quality components with low tolerances and performing calibration can improve overall accuracy.

Electrical Noise Reduction: Implementing noise reduction techniques in the design can minimize the impact of electrical noise on the ADC's accuracy.

Temperature Compensation: Temperature variations can affect measurement accuracy. Implementing temperature compensation techniques can help maintain accuracy across varying operating conditions.

## Conclusion:

In conclusion, the ohm meter project was a success, meeting the accuracy and independence criteria we set out to achieve. Selecting a 16-bit ADC offers high-resolution capabilities, but achieving 16-bit accuracy may be challenging. Understanding the trade-offs between resolution and accuracy and implementing strategies to enhance accuracy are vital for designing a precise ohm meter. Proper consideration of these factors will ensure that the ohm meter provides reliable and accurate measurements for various applications. The combination of a well-designed hardware setup, meticulous code implementation, and thorough testing contributed to the excellent results obtained. By reflecting on the project, we identified areas for improvement in future projects, such as planning testing schedules more efficiently and conducting breadboard testing before PCB assembly.

## Sources:

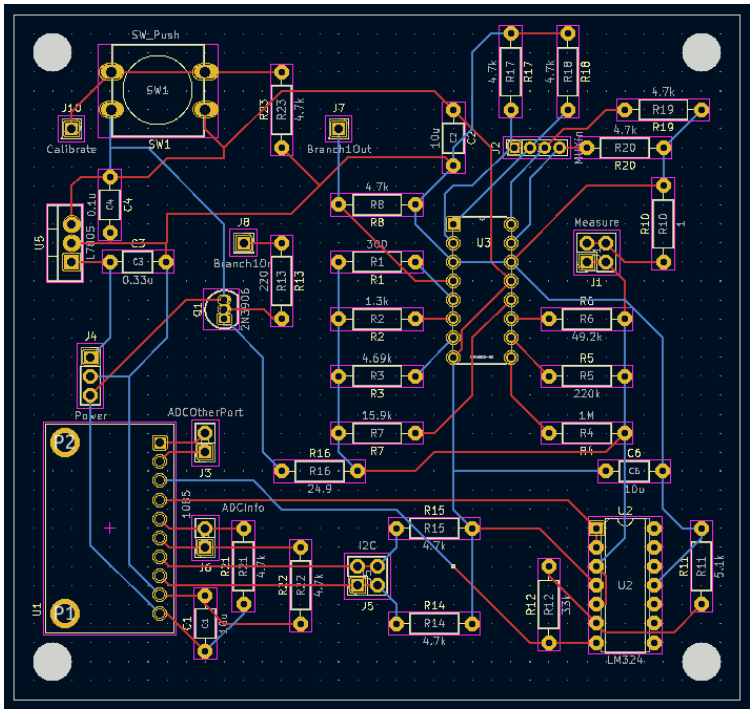

Arduino Official Website: https://www.arduino.cc/reference/en/

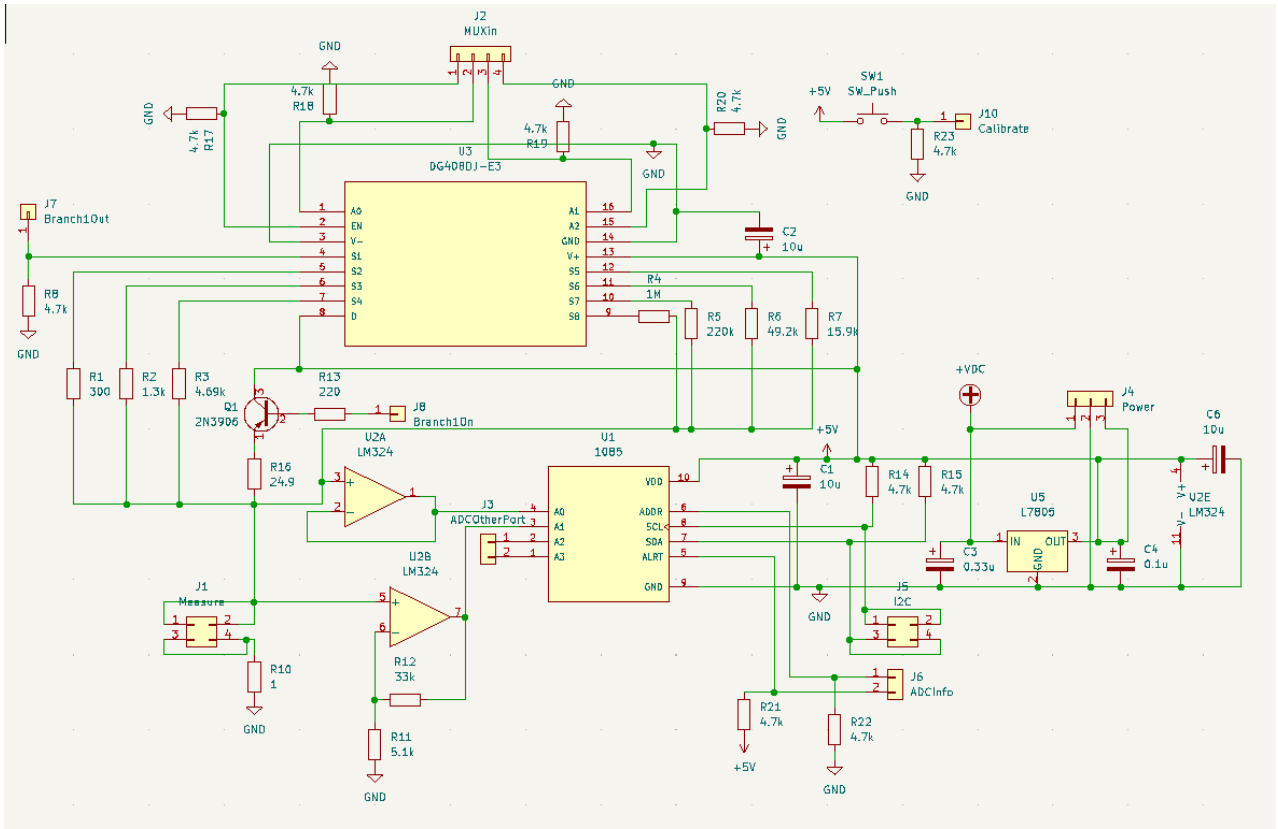
Arduino UNO Pinout: https://www.circuito.io/blog/arduino-uno-pinout/

AdaFruit ADC datasheet: https://cdn-shop.adafruit.com/datasheets/ads1115.pdf

Abstract:

Schematics, and PCB layout diagrams:



Photos of final design