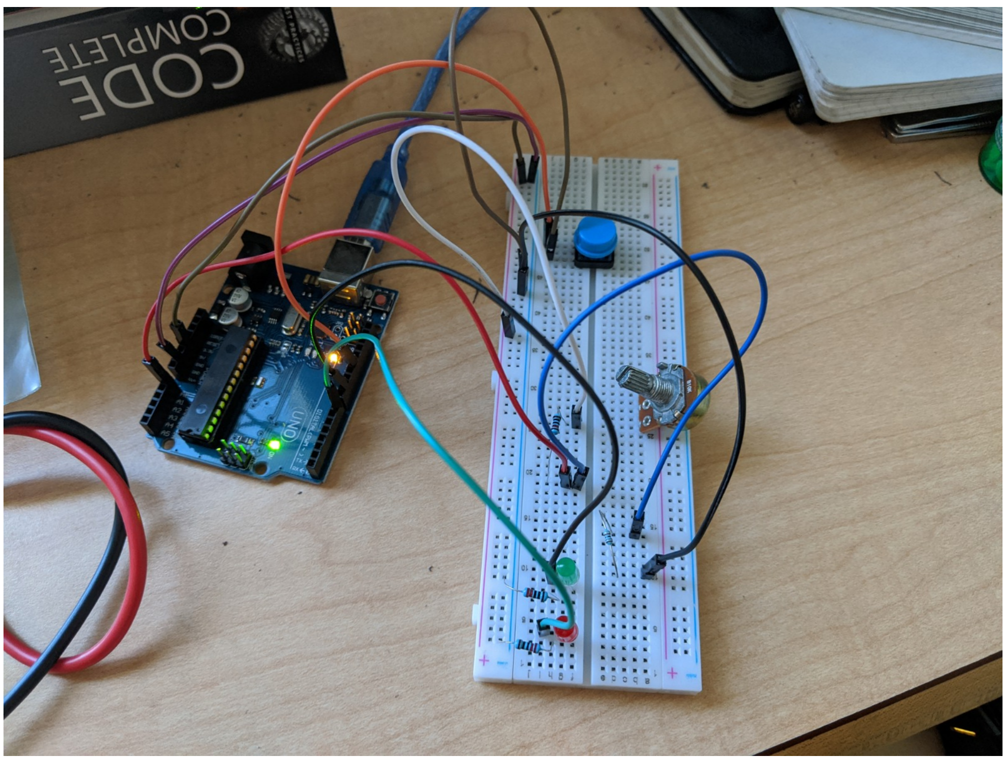
Instructor: Aiku Shintani

EE 143 Section 08
T 6:10-9PM

Experiment 6

Arduino Ohmmeter



Written by Astrid Yu
May 11, 2020

Introduction

In this experiment, an ohmmeter will be constructed from an Arduino. Code will be modified to improve the ohmmeter's measurements.

Analysis

a. An Arduino was connected to a computer and the IDE was configured to target the Arduino.

b. The circuit shown in Figure 1 was built.

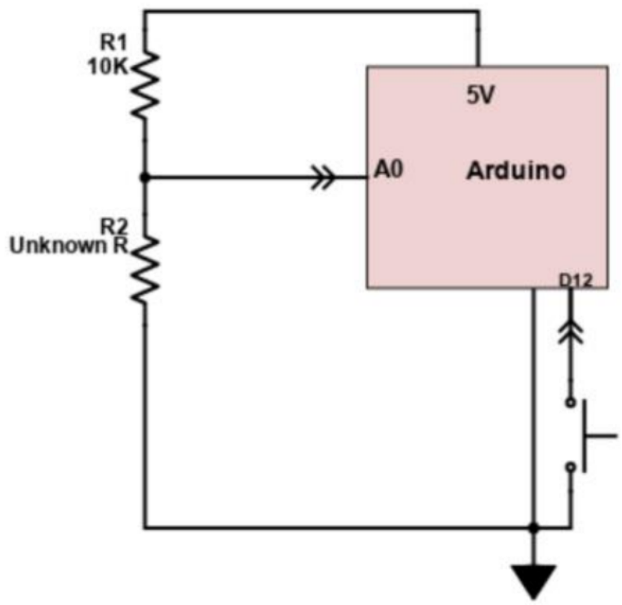


Figure 1: The ohmmeter circuit.

c. The provided code was uploaded to the Arduino.

d. The potentiometer was swept to various positions and the resistance was measured to confirm the code worked.

e. **Explain how the Arduino arrives at the R2 value displayed on the serial monitor.**

   The Arduino makes four measurements using the ADC so that the ADC can converge on a stable value. Then, it uses various transformations based on knowledge about the circuit in order to return the precise value.

f. A $4.3k\Omega \pm 1\%$ resistor was measured using a digital multimeter to have an actual resistance of $4.28k\Omega$.

g. The potentiometer was replaced with the $4.3k\Omega$ resistor.

h. The same resistor was measured using the Arduino Ohmmeter by pressing the button. The results from the serial console are seen in Figure 2

i. The outputted value was 4267.78Ω.

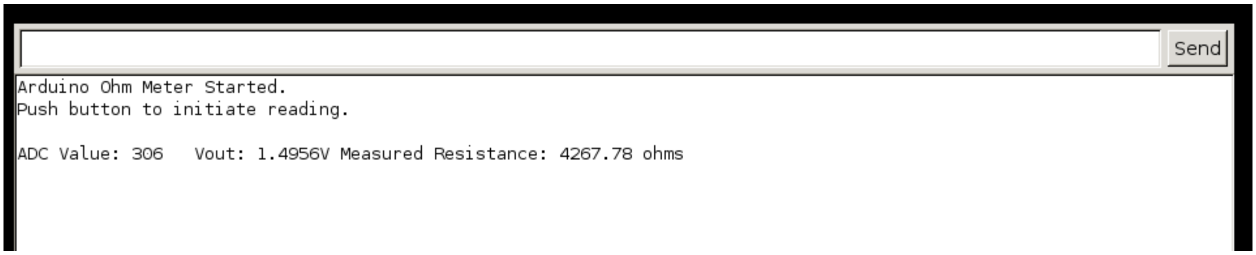


Figure 2: Arduino measurement of the resistor prior to calibration.

j. The error was calculated as follows:

$$\begin{aligned}\frac{\text{(arduino measured)} - \text{(actual)}}{\text{(actual)}} \cdot 100\% &= \frac{4267.78\Omega - 4.28k\Omega}{4.28k\Omega} \cdot 100\% \\ &= \mathbf{-0.303}\% \end{aligned} \tag{1}$$

k. The 5V ADC reference is likely to be not exactly 5V. Additionally, the $10k\Omega$ resistor has a non-zero tolerance on it. These variables could confound the measurement.

   The accuracy could be improved by compensating for the error by using this measurement as a calibration measurement and multiplying the final result by a constant k like so:

$$k \cdot \text{(arduino measured)} = \text{(actual)} \tag{2}$$

   Solving for k and plugging values in:

$$\begin{aligned} k &= \frac{\text{(actual)}}{\text{(arduino measured)}} \\ &= \frac{4.28k\Omega}{4267.78\Omega} \\ &= \mathbf{1.00286} \end{aligned} \tag{3}$$

l. The code was modified to use this scaling constant.

m. The recorded measurement was 4299.97Ω. The results from the serial console are seen in Figure 3

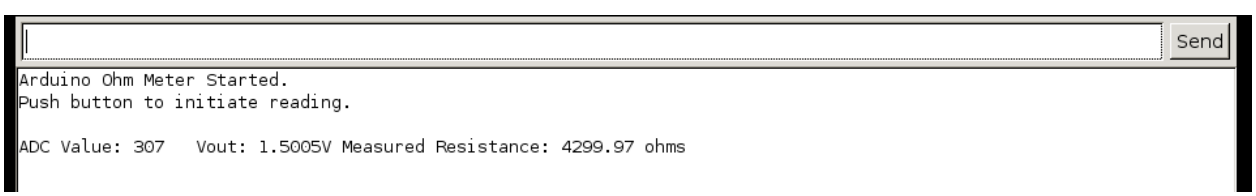


Figure 3: Arduino measurement of the resistor after calibration.

n. The error was calculated for this new measurement as follows:

$$\begin{aligned}\frac{\text{(arduino measured)} - \text{(actual)}}{\text{(actual)}} \cdot 100\% &= \frac{4299.97\Omega - 4.28k\Omega}{4.28k\Omega} \cdot 100\% \\ &= \mathbf{+0.4666}\% \end{aligned} \tag{4}$$

o. A red LED was added to the Arduino and the code was modified to turn the LED on when a short is detected.

p. A green LED was added to the Arduino and the code was modified to turn the LED on when an open is detected. See Figures 4, 5, and 6.

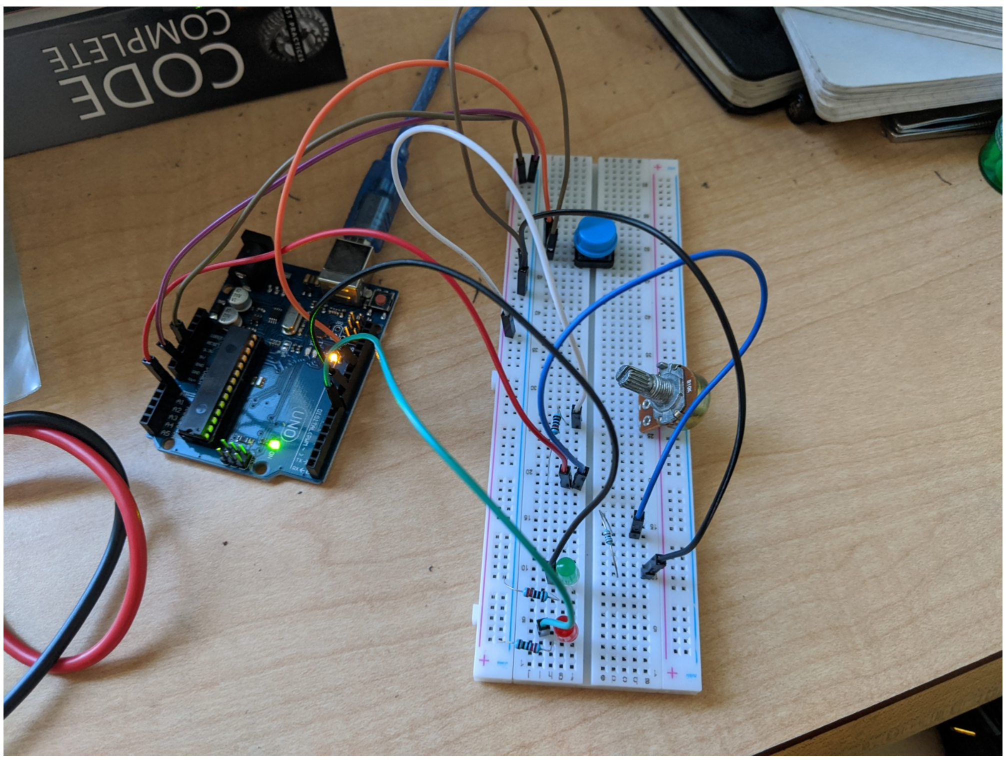


Figure 4: The circuit with a load between the two electrodes. Note that neither LED is lit.

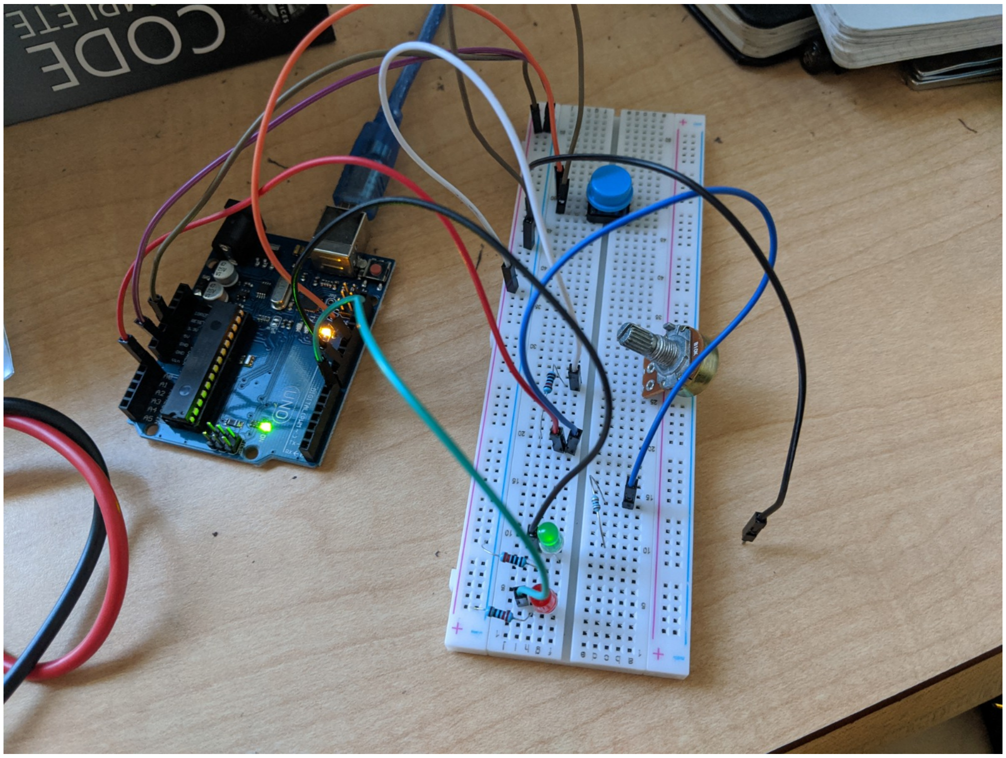


Figure 5: The circuit in open position. Note the green LED being lit, albeit dimly.

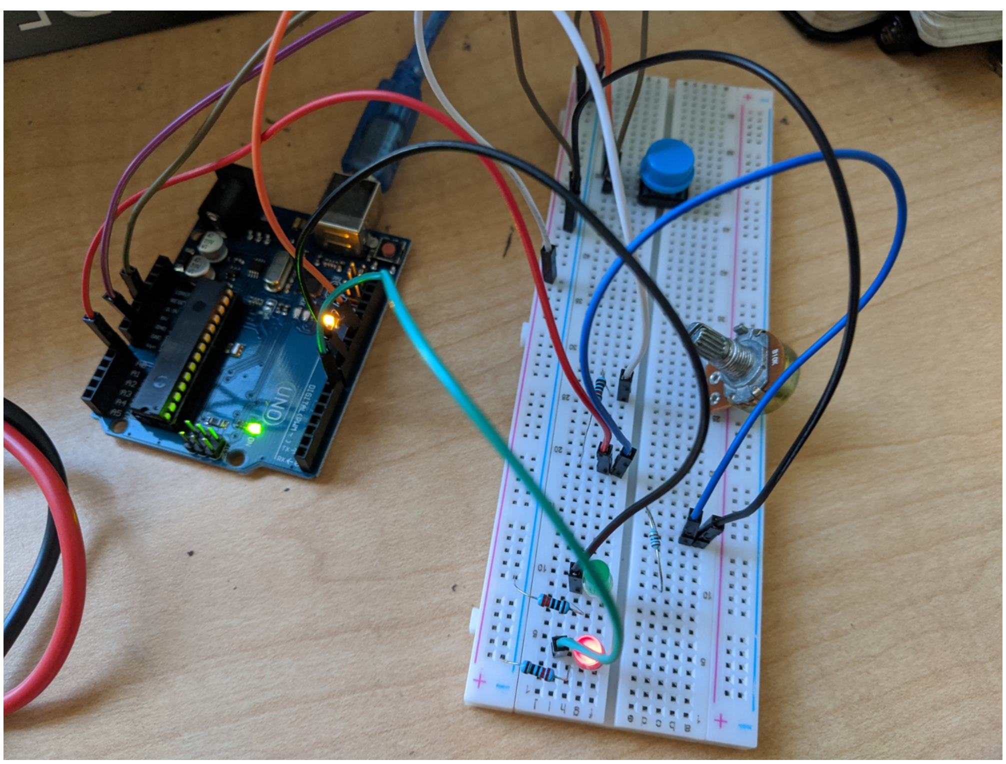


Figure 6: The circuit with a short. Note the red LED being lit.

Discussion Questions

1. The second time, a positive error was produced, and not only that, the error is even greater. This may be because the Arduino's readings are not only less accurate, but less precise. Only one calibration measurement was produced when it would probably better to make a few hundred. It is likely that the calibration measurement may have been a low outlier, causing the formula to overcompensate.

2. The source code begins on the next page.

Modified Source Code

```
#include <Bounce2.h>

#define LED_OPEN 9
#define LED_SHORT 7

const float Vin = 5.0;  //should start with Vin = 5.0
const float R1 = 10000; //should start with R1 = 10K
const int CIRCUIT = 0; //using analog input A0
const int BUTTON = 12; //Button to ground on pin 12
const int READ_DELAY_mS = 1; //delay between successive reads

//4.89 mV / ADC count, assuming 5V reference
const float VOLTS_PER_COUNT = Vin/1023.0;


Bounce Button = Bounce(); //Button object from Bounce library
float Vout; //voltage out of voltage divier, read by ADC
char VoutString[100]; //string representation of Vout
float R2; //"unknown" resistance
int ADCvalue; // value read from ADC

void setup() {
  // serial monitor used at 115200 bps
  Serial.begin(115200);
  // trigger button
  pinMode(BUTTON, INPUT_PULLUP);
  // using the Bounce2 library to debounce the button
  Button.attach(BUTTON,INPUT_PULLUP);
  Button.interval(25);
  // starting salutation
  Serial.println("Arduino Ohm Meter Started.");
  Serial.println("Push button to initiate reading.");
}

void loop() {
  // Is the button pushed?
  Button.update();
  bool ButtonPushed = !Button.read(); // invert logic to true = pushed

  // if the button is pushed, generate a reading
  if (ButtonPushed) {
    //read the ADC four times to get a stable reading
    for (int i=0; i < 4; i++) {
```

```
      ADCvalue = analogRead(CIRCUIT);
      delay(READ_DELAY_mS);
    }
    //calculate the voltage that was read
    Vout = (float)ADCvalue * VOLTS_PER_COUNT;
    // calculate the resistance
    R2 = 1.00286 * (Vout*R1)/(Vin-Vout);
    // print the results to the serial monitor
    Serial.print("ADC Value: ");
    Serial.print(ADCvalue);
    Serial.print("   Vout: ");
    dtostrf(Vout, 1,4, VoutString);
    Serial.print(VoutString);
    Serial.print("V");
    Serial.print(" Measured Resistance: ");
    Serial.print(R2);
    Serial.println(" ohms");
    // wait for the button to be released before proceeding
  }

  int continuityMeasurement = analogRead(CIRCUIT);
  bool openState = 0;
  bool shortState = 0;
  if (continuityMeasurement == 1023) {
    openState = 1;
  } else if (continuityMeasurement == 0) {
    shortState = 1;
  }
  digitalWrite(LED_OPEN, openState);
  digitalWrite(LED_SHORT, shortState);

  while (ButtonPushed) {
    Button.update();
    ButtonPushed = !Button.read();
  }
}
```

Conclusion

In this experiment, a digital ohmmeter was successfully constructed and tested. Attempts were made to improve the readings were not successful, likely due to a small sample size. In a future experiment or in a production environment, more measurements would need to be made and analyzed to ensure accuracy and precision together.