An Arduino Uno Ohm Meter

Dr. Matthew Riehl

August 31, 2021

1 Introduction

This week, you will build an ohmmeter (Ω-meter) to measure the resistance of individual resistors and combinations of resisters. The lab can be open-ended and you are encouraged to find ways to improve the circuit and code to make a more versatile instrument.

2 Materials

The materials below were assembled last time to display a brief message from the Arduino on the LCD. Check to ensure that the board is still assembled and that the display still works. If it does not, your first job is to fix it.

- Computer with Arduino IDE installed.
- Arduino Uno board and power cable
- Half-size breadboard
- LCD display module
- 220 Ω resistor (Ω is Ohm)
- 10 kΩ potentiometer
- Jumper wires

The materials here are the new parts needed for completion of this weeks project.

- Jumper wires
- Resistors (variety)
- Commercial multi-meter

3 Background

Resistors are used control the current and voltage that flows through a circuit and are commercially available with resistances from a few ohms to gigaohms (1×10^9 Ω). They are also manufactured to tolerances from $\pm 0.05\%$ to $\pm 10\%$ because, when choosing a resistor for a circuit, sometimes 'close is good enough' and at other times the exact resistance must be known. Most of the resistors you will encounter have a stated resistance of $\pm 5\%$, so the error is considerable. If you need to know the resistance in a circuit or through a collection of resistors, an ohmmeter is a useful tool to have in your shop.

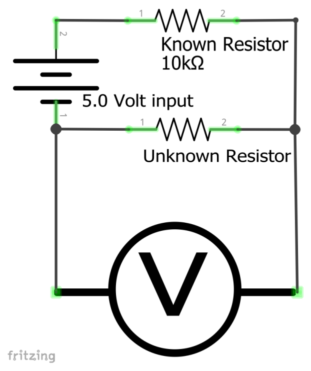

Figure 1: Voltage Divider

The ohmmeter works by building a 'voltage divider' circuit as shown in Figure 1. The voltage measured by the voltmeter can be calculated as follows: The battery supplies some amount of current that passes first through Resistor 1 and then Resistor 2. Keeping in mind Ohm's law: $V = IR$, we can see that if the current through both resistors is the same (as it must be), then,

$$I = \frac{V}{R} = \frac{V_{Resistor\ 1}}{R_{Resistor\ 1}} = \frac{V_{Resistor\ 2}}{R_{Resistor\ 2}} \tag{1}$$

Recall that $V = \dfrac{EPE}{q}$ and the electrical potential energy is consumed, or lost, as the charges flow through the circuit. There is a voltage drop through each leg of the circuit (through each resistor) which can be measured with a voltmeter. If the voltage supplied to the circuit by the battery is 5 V, the measured voltage (the drop across Resistor 2) will always be less than 5 volts, but how much less depends on the resistances of both resistors. In the simple case where $R_1 = R_2$, the voltmeter will read 2.5 Volts, but as R_2 becomes smaller, the voltage drop across R_2 becomes larger. From this observation and Ohm's law, we conclude that a smaller resistance in a circuit allows more current to flow.

For the circuit shown in Figure 1, we can write:

$$V = V_1 + V_2 = IR_1 + IR_2 = I(R_1 + R_2) = IR_{equivalent} \tag{2}$$

where $R_{equivalent}$ is the sum of the resistors or the equivalent resistance. Since the current is the same through each leg of the circuit, we can make the substitution $I = \dfrac{V_2}{R_2}$ and rearrange:

$$V = \frac{V_2}{R_2}(R_1 + R_2) \tag{3}$$

$$V_2 = \left(\frac{V}{R_1 + R_2}\right) R_2 \tag{4}$$

Equation 4 is the general equation for a voltage divider. A power source may supply more voltage than is needed for a particular application and the voltage can be reduced by choosing a combination of resistors such that V_2 is the desired voltage.

For our purposes today, however, we will measure V_2 and use our knowledge of V and R_1 to determine the value of R_2. Equation 3 can be rearranged to yield

$$R_2 = \frac{R_1 V_2}{V - V_2} \tag{5}$$

and this is the equation we will program into the micro-controller.

4 Arduino Uno Wiring

The LCD display you built last time will not be changed. You will add a few wires and one resistor to the breadboard to complete the voltage divider circuit. The 5 volt potential will be supplied by the Adafruit 5 V pin. The circuit will be completed by connecting the voltage divider to the ground in the power strip. A resistor of known resistance will be added to the circuit as shown. Two leads may be used to connect to the unknown resistor or resistors if desired. The green wire is used to measure the voltage between R_2 and the ground, and it is connected to the A5 pin on the Arduino Uno. The A0 - A5 pins on the Adafruit are analog pins. The Arduino then digitizes the voltage – it assigns a digital value where 0 equals 0 volts and 1024 equals 5.0 volts ($1024 = 2^{10}$); you will see in the code that the value taken from A5 must be multiplied by $\frac{5.0}{1024}$ to yield a voltage. See circuitbasics.com/arduino-ohm-meter/ for more information.

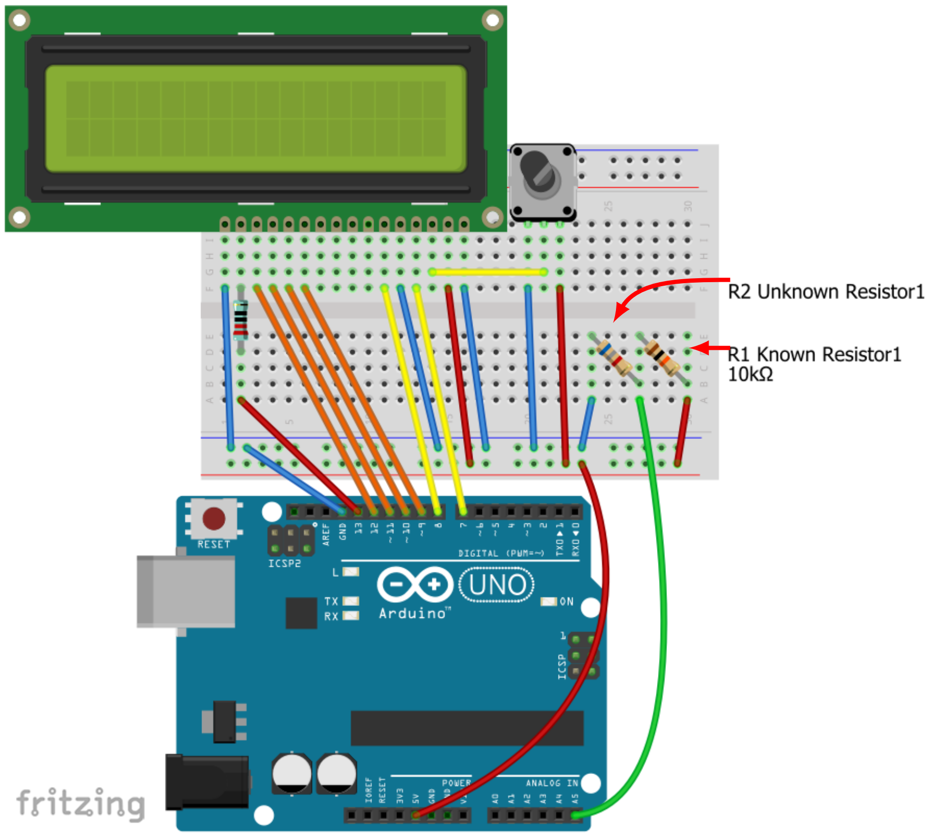

Figure 2: Pin-out Diagram for the Arduino Uno Ohmmeter.

The unknown resistors can be added to the circuit as convenient – the resistor may be connected directly to the breadboard or a second breadboard may be used.

5 C++ Code for the Arduino Uno

The code for this weeks lab is reproduced below, but is also included in a separate .ino file. Note that the value of the known resistor is set to $10,000\ \Omega$ in this code (R1=10000;). You will change this to whatever known resistor you are using. Remember that you must upload the new code to the Arduino Uno before it takes effect.

As supplied, the code will inform you of the resistance coded in for the known resistor and then display the resistance measured for the unknown resistor and the voltage drop across the unknown resistor. If either resistor is absent or poorly matched to the other resistor (resulting in a large error in the calculated resistance), a message is sent to the LCD and then the display shuts off until the error is fixed.

```
/*

This code is included with the files for Arduino Lab Ohm_meter by
Matthew Riehl and is in the public domain.  I have only modified the
code to better support the laboratory exercise.

Note that the pin assignments for the LCD are NOT the same as found on many Arduino
projects.  They have been moved around to make the wiring more intuitive
and easier to troubleshoot.  See the Arduino Hello World exercise for
more information regarding the LCD display wiring.

The Ohmmeter circuit uses a voltage divider.  A five volt potential flows
through two resistors in series.  The first resistor has a known resistance
and is assigned R1 in the code.  (You MUST change R1 in the code to match the
resistance being used.)  The resistance of the second (unknown) resistor
is calculated from Equation 5 in the handout.  The LCD displays the voltage
drop across R2 (the unknown resististance) and the calculated resistance.
The accuracy of the measurement drops at voltages approaching 0 (unknown
resistors with resistances much higher than R1) and at voltages approaching
5 volts (unknown resistors with resistances much lower than R1).

If either R1 or R2 are absent or out of range, the display shuts off


This example code is in the public domain.

*/

// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define backlightPin 13         // This allows the user to turn the back light on and off.

int Vin=5;        //voltage at 5V pin of arduino
float Vout=0;     //voltage at A5 pin of arduino
float R1=527;    //value of known resistance
int known=0;
float R2=0;
int a2d_data=0;
```

```
float buffer=0;

void setup()
{
lcd.begin(16,2);
pinMode(backlightPin, OUTPUT);
digitalWrite(backlightPin, HIGH);
lcd.setCursor(0,0);
lcd.print("Ohmmeter!");
delay(3000);
known = int(R1);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("R1 Resistance");
lcd.setCursor(0,1);
lcd.print(known);
lcd.print(" Ohms");
delay(5000);
lcd.clear();
}

void loop()
{
a2d_data=analogRead(A5);  // The digital voltage equivalent read across R1 (between 0 (0 volts) and 102

// This loop will calculate the resistance of R2 if the voltage drop across
// R1 is between about 0.5 volts and 4.5 volts.  Otherwise it will complain.
// This range can be changed -- a smaller range might give more accurate values
// for R2, but will require switching out R1 more often.
if(a2d_data <= 924 and a2d_data >= 100)   // Set the range here -- no need to change elsewhere.
{
digitalWrite(backlightPin, HIGH);  // Turn lcd backlight ON
buffer=a2d_data*Vin;
Vout=(buffer)/1024.0;
buffer=Vout/(Vin-Vout);
R2=R1*buffer;

lcd.setCursor(0,0);
lcd.print("R2 Voltage: ");
lcd.print(5-Vout);

lcd.setCursor(0,1);
lcd.print("R2 (ohms) = ");
lcd.print(R2);

delay(500);
}

// If R1 (the known resistor) is absent, the backlight turns off

else if(a2d_data <= 512)
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("R1 missing");
lcd.setCursor(0,1);
lcd.print("or R1 >> R2");

delay(5000);
while(a2d_data < 100)
{
a2d_data=analogRead(A5);
digitalWrite(backlightPin, LOW);  // Turn backlight off
lcd.clear();

delay(500);
}
}

// If there is no resistor to measure, the backlight turns off
// untill a resister is placed in the circuit.
else
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("R2 missing");
lcd.setCursor(0,1);
lcd.print("or R1 << R2");

delay(5000);
while(a2d_data > 924)
{
a2d_data=analogRead(A5);
digitalWrite(backlightPin, LOW);
lcd.clear();

delay(500);
}
}
}
```

When you have a working ohmmeter, ask your lab instructor for permission before continuing.

6 Errors in Measuring Resistance

Keep a record of your work and calculations on a separate page that can be turned in with your report.

1. Select a resistor to be your known resistor R_1 and insert it into the circuit as shown in the pinout diagram (Figure 2). Change the value of R1 in the C++ code to the resistance of R_1 and upload this to the Arduino Uno. Be sure to record the value of R1 in your notes. You may wish to verify the resistance of R_1 with a commercial ohmmeter.

2. Select a variety of resistors (at least five) to be R2 and use the ohmmeter to measure the resistance across each one. For each resistor, record the resistance as indicated by the colored bands and record the resistance as determined by your ohmmeter.
3. Calculate the percent error between the two resistance values.

$$\%\ Error = \frac{\text{Expected Resistance - Measured Resistance}}{\text{Expected Resistance}} \times 100 \tag{6}$$

4. Repeat steps 1 - 3 with a different R_1. Choose resistors for R_1 that differ by orders of magnitude, such as $100\ \Omega$, $1000\ \Omega$, and $100,000\ \Omega$.

You should find that the errors are smaller when R_2 has about the same resistance as R_1. If this is not the case, inform the lab instructor.

7 Resistors in Series and in Parallel

Again, please keep a record of your work and calculations on a separate page.

In this section, you will investigate the resistance of a circuit that contains multiple resistors in series or in parallel. The Arduino Uno will measure the resistance of the circuit and you will compare this to the expected resistance of the circuit, calculated from the known resistances of each resistor.

As we saw earlier, the equivalent resistance of two or more resistors in series is the sum of the individual resistances.

$$R_{series} = R_1 + R_2 + \cdots = \sum_i R_i \tag{7}$$

When resistors are in parallel, however, the current is divided between the resistors with the smallest resistor receiving the greatest current (the path of least resistance). It will be shown in class that the equivalent resistance of resistors in parallel is:

$$\frac{1}{R_{parallel}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots = \sum_1 \frac{1}{R_i} \tag{8}$$

1. Construct circuits that contain at least two or three resistors in series (Figure 3) and measure the resistance with your ohmmeter. If necessary, replace your R_1 with a resistor suitable for the resistance you are measuring.
2. Draw each circuit, label each resistor with the known resistance, and calculated the expected resistance of the circuit.

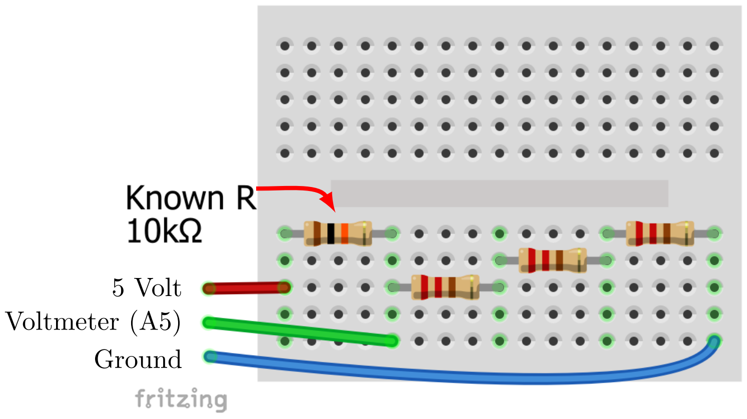

Figure 3: Three resistors in series circuit diagram

3. Calculate the percent error.
4. Repeat steps 1-3 with circuits that contain at least two or three resistors in parallel (Figure 4).
5. Repeat steps 1-3 with circuits that contain at least three or four resistors with both parallel and series components (Figure 5).

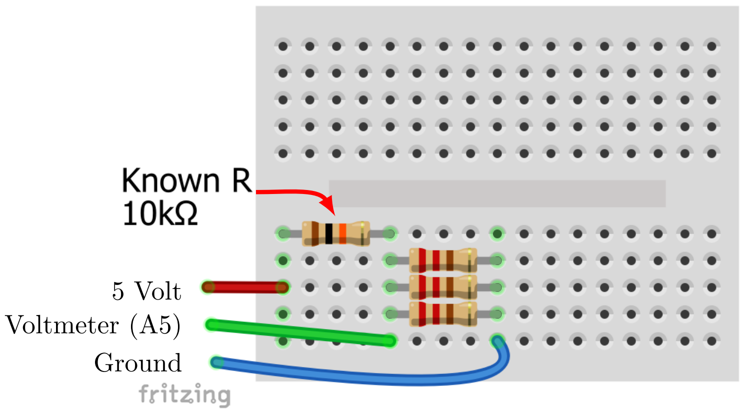

Figure 4: Three resistors in parallel circuit diagram

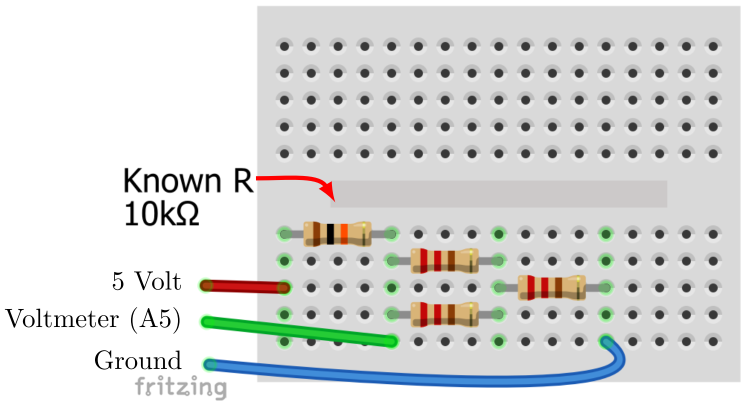

Figure 5: Two resistors in parallel, in series with a third

8 Prelab

Part 1: Below, you will find thirteen resistors of varying values (Figure 6). Use the resistor color chart to determine the ohm value of each resistor. (These are all ±5%)

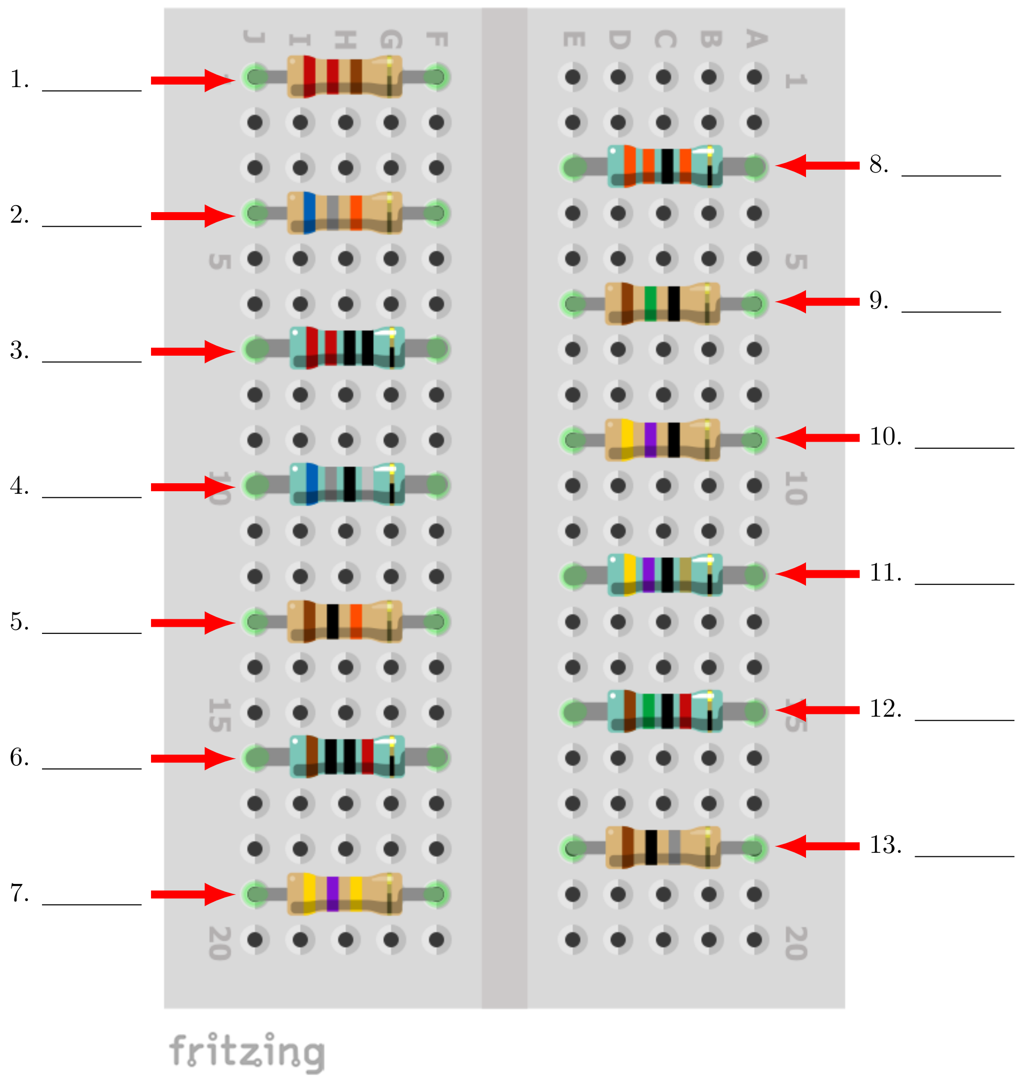

Figure 6: Thirteen unidentified resistors

Part 2: Using your best artistic skills and colored pens/pencils, draw the following resistors with the correct bands: i. 810 $\Omega \pm 0.5\%$, ii. 80×10^8 $\Omega \pm 0.10\%$, iii. 21 $\Omega \pm 2\%$. The following use 5-Band Codes: iv. 22200 $\Omega \pm 5\%$, v. 764×10^9 $\Omega \pm 0.25\%$.

Color	1st Band	2nd Band	3rd Band	Multiplier	Tolerance
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	±1%
Red	2	2	2	100Ω	±2%
Orange	3	3	3	1kΩ	
Yellow	4	4	4	10kΩ	
Green	5	5	5	100kΩ	±0.5%
Blue	6	6	6	1MΩ	±0.25%
Violet	7	7	7	10MΩ	±0.10%
Grey	8	8	8	100MΩ	±0.05%
White	9	9	9	1GΩ	
Gold				0.1Ω	±5%
Silver				0.01Ω	±10%

Figure 7: Color banding codes for resistors

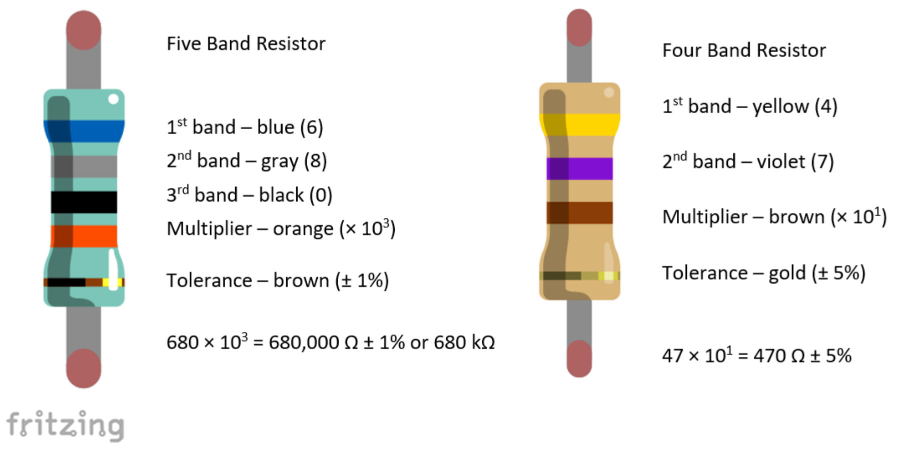

Figure 8: Examples of a 5-band and 4-band resistor