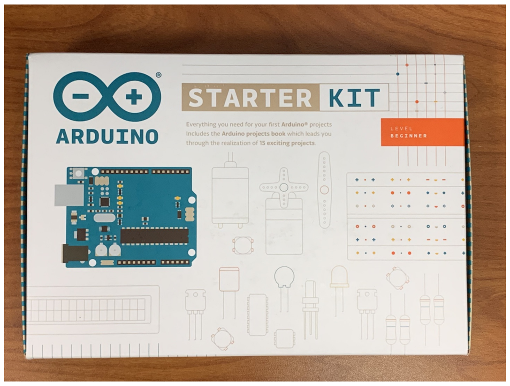
Nima Alamatsaz
Antje Ihlefeld

LAB 1

Introduction to Arduino, Breadboard and Resistors

You should have received two kits: Arduino starter kit and an extra lab kit. In lab 1 we'll be mostly using the contents of Arduino kit, and only one component from the extra lab kit.

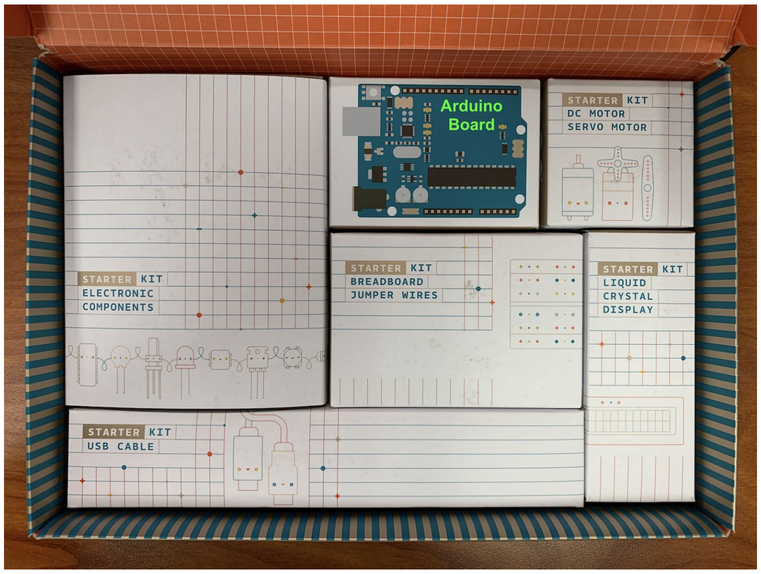


Arduino starter kit

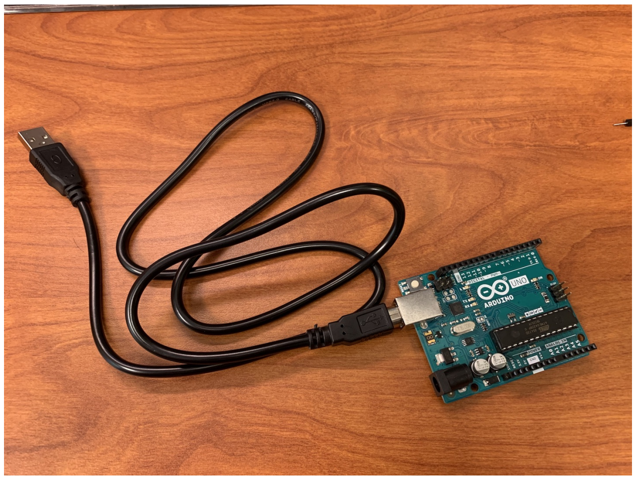
Extra lab kit

1 Arduino

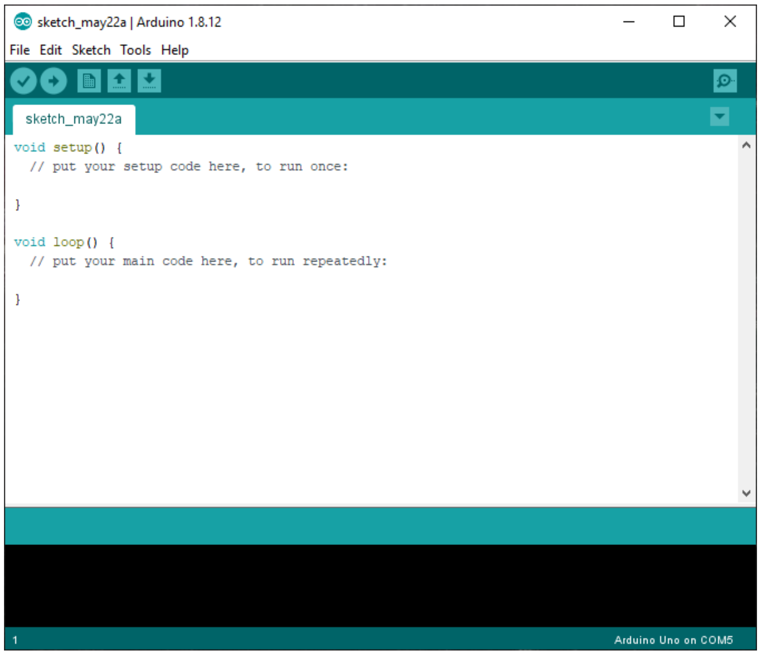
Arduino is an open-source electronics platform that allows easy and rapid development of projects, ranging from hobby electronics to actual devices including biomedical instruments. At the heart of Arduino is a programmable microcontroller which acts as a tiny computer capable of running custom software. This software lets us monitor and control electronics that are connected to the Arduino board. Before getting started with Arduino, we have to download the appropriate software. Go to the following link and download Arduino IDE: www.arduino.cc/en/Main/Software


Arduino Board
STARTER KIT DC MOTOR SERVO MOTOR
STARTER KIT ELECTRONIC COMPONENTS
STARTER KIT BREADBOARD JUMPER WIRES
STARTER KIT LIQUID CRYSTAL DISPLAY
STARTER KIT USB CABLE

Open up your Arduino kit and remove the Arduino board and USB cable from their boxes, then, connect them together and into the USB port of your computer.

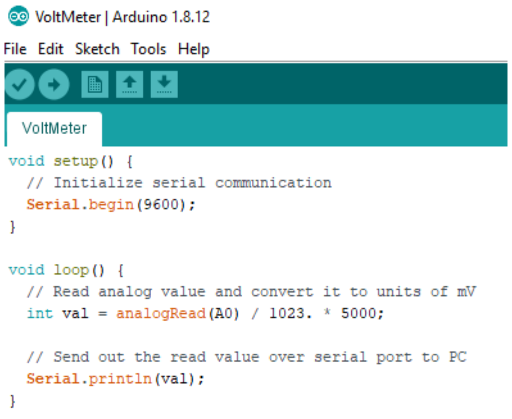


Now, launch the Arduino software you installed earlier and wait out the loading screen. You should be directly taken to a screen that looks like the picture below. This is the template code for an empty software. Although it might not look like much, but this screen provides you with almost infinite possibilities! Here you can write code for programs as simple as blinking LEDs, timed traffic lights, or even AI-based prediction of stroke in a wearable ECG device! These programs are written in the C++ language, which is a powerful language which most operating systems (including Windows, macOS and Linux) are written in.



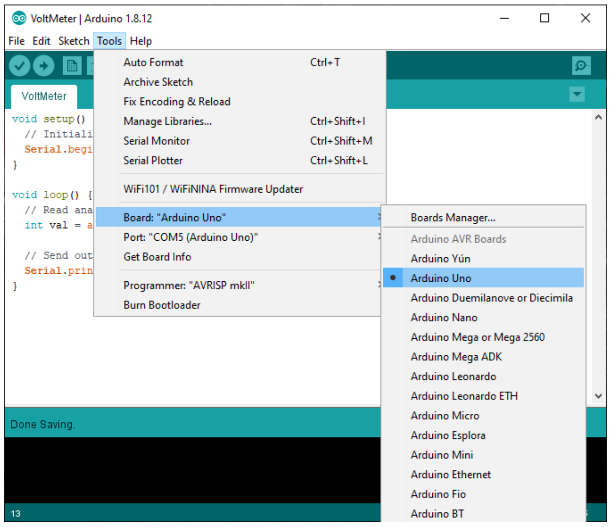
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

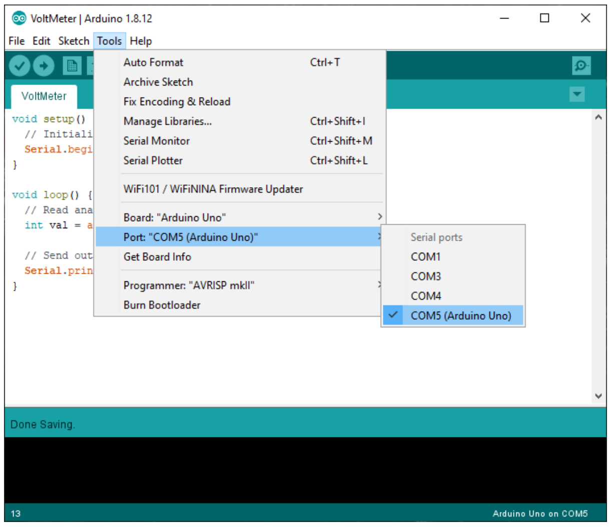
Normally in lab we would have access to equipment such as multimeter or oscilloscope to help us with electrical measurements. However, for our at-home labs we are going to write a simple but very useful program to turn Arduino into an oscilloscope, with only 3 lines of code! The picture on right shows the code that you have to write. Pay extra attention to character cases because C++ is a case-sensitive language. Lines starting with two forward slashes (//) are not a part of the program, they are just comment lines that describe what the following portion of code is doing. Feel free to skip them in your own code. If interested, you may want to take a look over an in depth tutorial on Arduino oscilloscope. Save your project with an appropriate name.



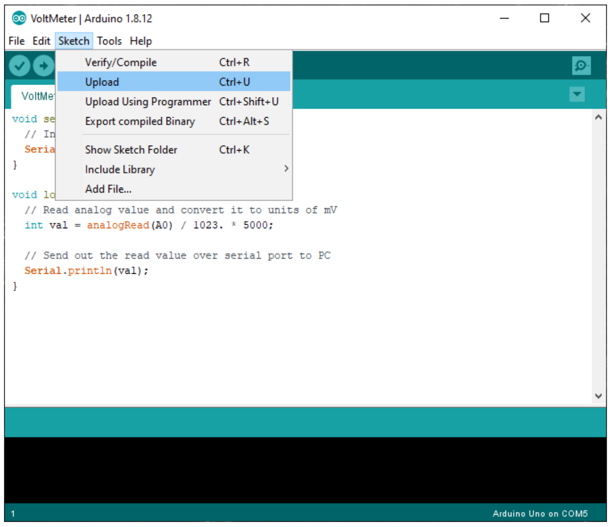
```
void setup() {
  // Initialize serial communication
  Serial.begin(9600);
}

void loop() {
  // Read analog value and convert it to units of mV
  int val = analogRead(A0) / 1023. * 5000;

  // Send out the read value over serial port to PC
  Serial.println(val);
}
```

Before uploading the oscilloscope program onto Arduino, we need to let the software know what kind of Arduino we are using and which port is it connected to. Open the Tools menu and try to find “Arduino Uno” under the Board submenu. Then, open the Port submenu and select the COM port that has written “Arduino Uno” in front of it.

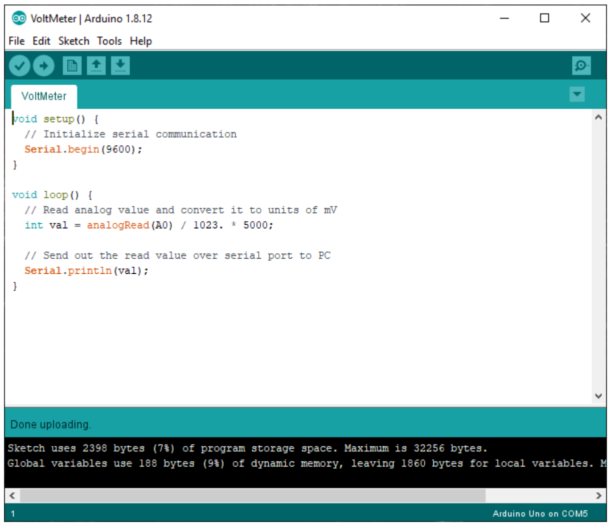



VoltMeter | Arduino 1.8.12
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM5 (Arduino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM3
COM4
COM5 (Arduino Uno)
Done Saving.
13
Arduino Uno on COM5

From the Sketch menu select Upload to transfer the program to the Arduino board. After about a few seconds you should see a message at the bottom saying “Done uploading”. If instead you are given an error, first make sure Arduino is properly connected to your computer via the USB cable, and that the software is correctly recognizing it (look for “Arduino Uno on COM5” or another port number at the bottom right). Also double check your code for any small mistakes, remember in the C++ language every character is important!


VoltMeter | Arduino 1.8.12
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Arduino Uno on COM5

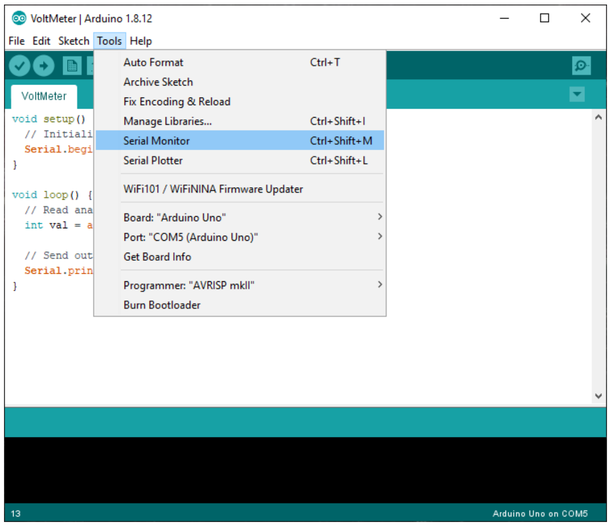
```
void se
  // In
  Seria
}

void lo
  // Read analog value and convert it to units of mV
  int val = analogRead(A0) / 1023. * 5000;

  // Send out the read value over serial port to PC
  Serial.println(val);
}
```

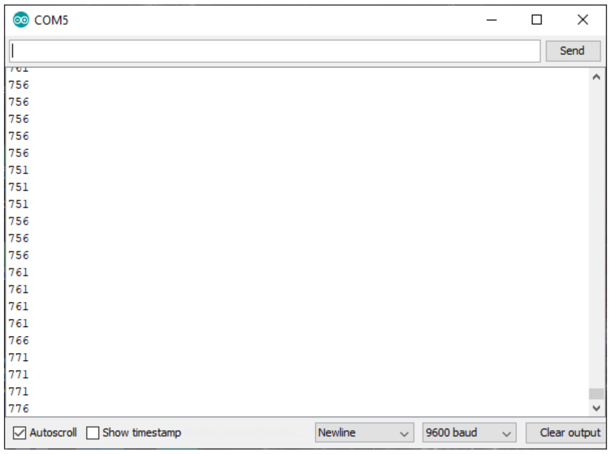


```
void setup() {
  // Initialize serial communication
  Serial.begin(9600);
}

void loop() {
  // Read analog value and convert it to units of mV
  int val = analogRead(A0) / 1023. * 5000;

  // Send out the read value over serial port to PC
  Serial.println(val);
}
```

```
Done uploading.
Sketch uses 2398 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 188 bytes (9%) of dynamic memory, leaving 1860 bytes for local variables. M
```

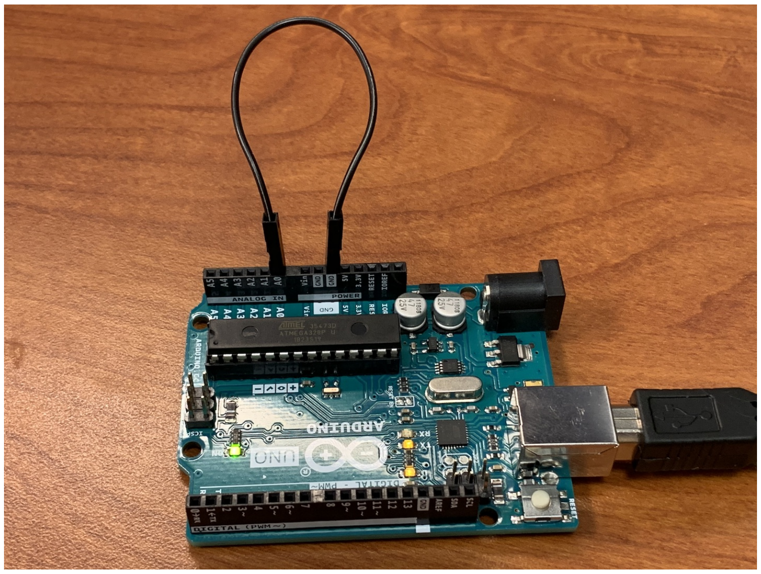
To test our freshly programmed oscilloscope, open Serial Monitor under the Tools menu, and on the bottom right of the opened window select "9600 baud" from the drop-down list. If things work as they should, you'd see a bunch of seemingly random numbers that keep changing quickly. These numbers are the voltage that Arduino is measuring at pin A0 in units of millivolts. Since A0 is not connected to anything at the moment, these numbers mostly show the bias voltage of the internal circuitry (what?!) plus the electrical noise of the environment.


VoltMeter | Arduino 1.8.12
File Edit Sketch Tools Help
VoltMeter
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM5 (Arduino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
13
Arduino Uno on COM5



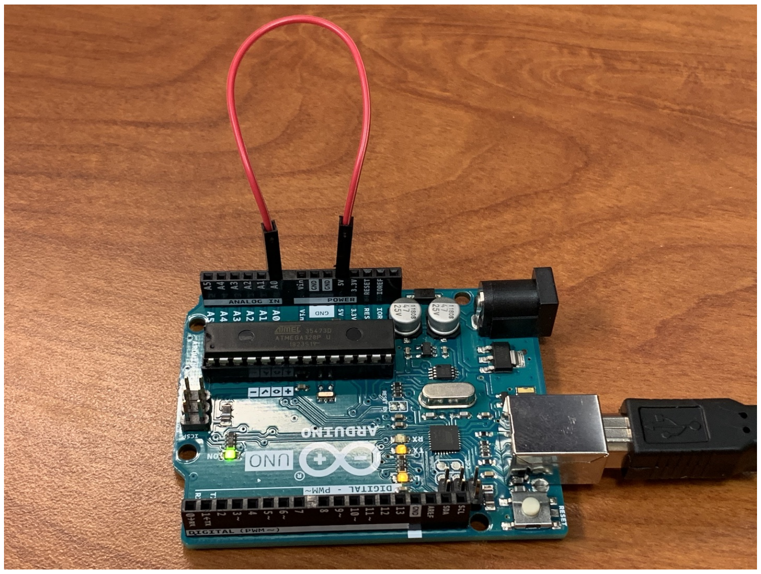
```
756
756
756
756
756
751
751
751
756
756
756
761
761
761
761
766
771
771
771
776
```

1.1. Now we are ready to take some measurements! Take out some jumper wires either from the Arduino kit (in the Breadboard and Jumper Wires box) or from the extra lab kit. The wires look different but they serve the same purpose. Note that the color of the wires do not matter, they all conduct electricity the same way. However, it is always a good idea to follow the conventional color code for GND being black or blue, and Vcc (+5v) being red. Use one of the wires to connect GND to A0.



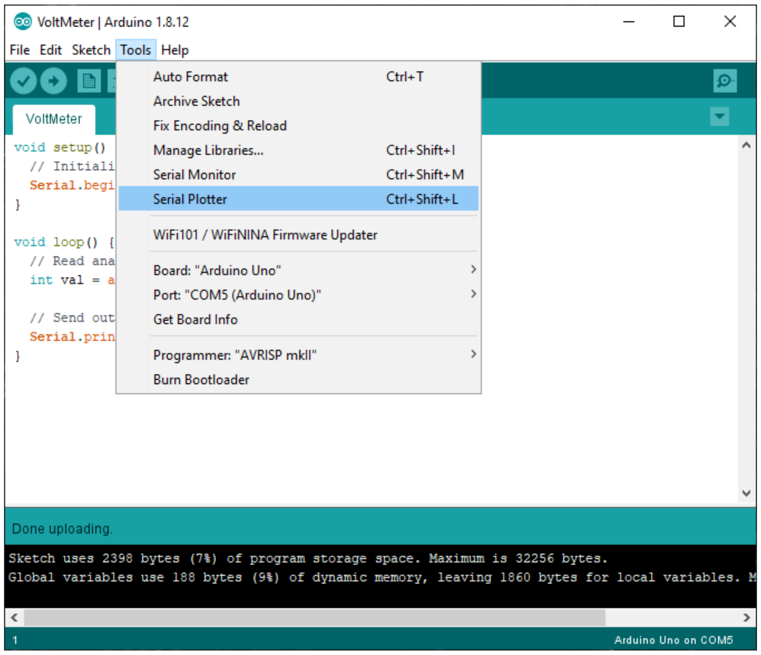
What value does Serial Monitor show when A0 is connected to GND? What is the unit for this value? Snap a screenshot of Serial Monitor for you report.

1.2. To further verify how well our oscilloscope is working, first connect A0 to the 5V pin, and then connect it to the 3.3V pin. When voltage measurements on Serial Monitor are changing quickly, since the values are relatively close to each other, reading any of them would suffice.



What value does Serial Monitor show when A0 is connected to 5V and 3.3V? Snap screenshots of Serial Monitor for you report.

1.3. So far we have seen Arduino act mostly as a multimeter for reading voltages. For seeing an actual live plot of the readings like an oscilloscope, close Serial Monitor and instead open Serial Plotter from the Tools menu. Remember to select “9600 baud” from the drop-down list on the bottom left corner.


VoltMeter | Arduino 1.8.12
File
Edit
Sketch
Tools
Help
VoltMeter
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM5 (Arduino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
1
Arduino Uno on COM5

```
void setup()
  // Initiali
  Serial.begi
}

void loop() {
  // Read ana
  int val = a

  // Send out
  Serial.prin
}
```

Done uploading.

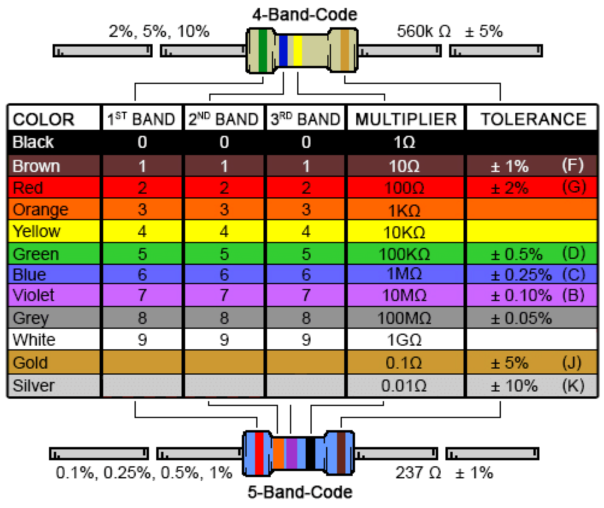
```
Sketch uses 2398 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 188 bytes (9%) of dynamic memory, leaving 1860 bytes for local variables. M
```

Snap a screenshot of Serial Plotter when A0 is connected to 3.3V.

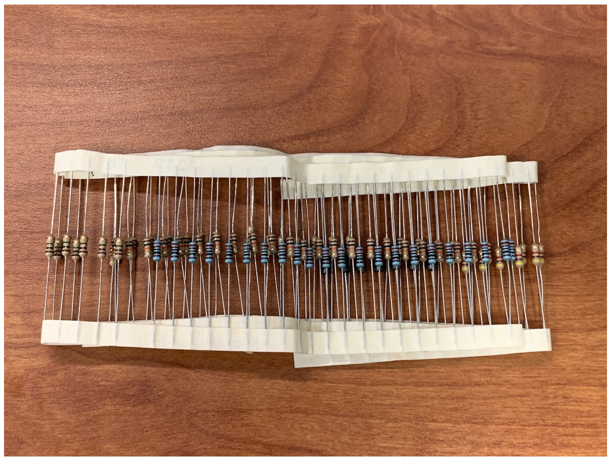
We won’t be using Serial Plotter anymore in lab 1 since we will mostly be dealing with constant voltages. In lab 2 we take full advantage of Serial Plotter for observing time-varying voltages of RC circuits.

2 Resistors

In this section we will verify our knowledge of resistors by using the color code to read theoretical resistance values, and come up with a simple circuit to calculate the actual resistance value without a multimeter. If you happen to have a multimeter at home, or would like to invest in one, I encourage you to verify all measurements with it and write them in your report. Doing so is optional.


4-Band-Code
2%, 5%, 10%
560k Ω ± 5%
0.1%, 0.25%, 0.5%, 1%
237 Ω ± 1%
5-Band-Code

COLOR	1ST BAND	2ND BAND	3RD BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8	100MΩ	± 0.05%
White	9	9	9	1GΩ	
Gold				0.1Ω	± 5% (J)
Silver				0.01Ω	± 10% (K)



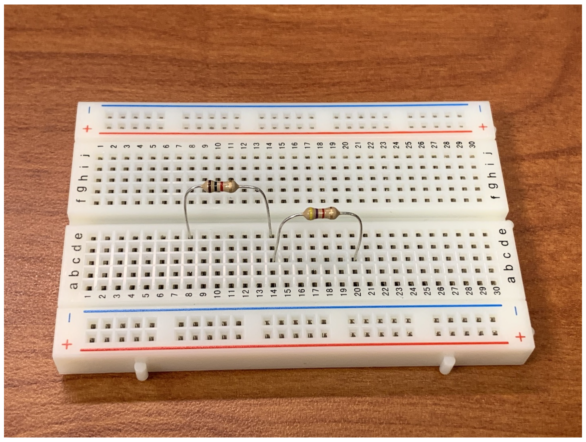
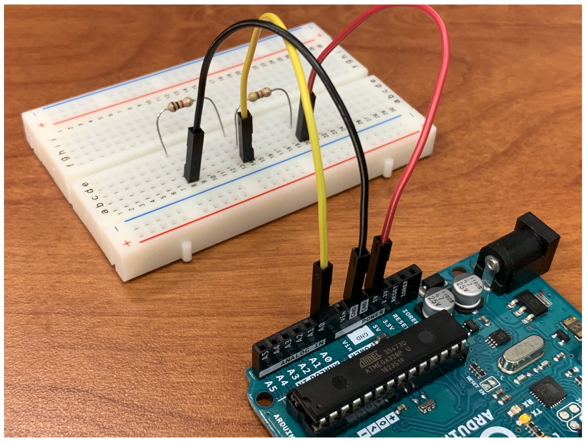
2.1. Open up the Electronic Components box from your Arduino kit and take out the resistor strip shown in the picture above. Use the resistor color table to find a $1k\Omega$ and a $4.7k\Omega$ resistor.

Write down the band colors for a $1k\Omega$ and a $4.7k\Omega$ resistor, and snap a photo of the two resistors you found for the report.

2.2. Try reading the tolerance for both resistors.

What color is the last band? What is the tolerance percentage? In what range should the actual resistance value for each resistor fall?

2.3. To verify our theoretical resistance readings we are going to use a voltage divider circuit. Take out a breadboard either from the Arduino kit or the extra lab kit. Place the two resistors on breadboard in series as shown in the picture.

Make sure to push the resistors all the way into the breadboard so that they are not loose and make perfect electrical contact, and pay extra attention to put the $1k\Omega$ resistor on the left side and $4.7k\Omega$ on the right. Using three jumper wires, connect the left pin of the $1k\Omega$ resistor to GND, the right ping of the $4.7k\Omega$ resistor to 5V, and the middle pins to A0. Use the picture in the previous page as your guide.

Draw the full circuit diagram of what you have assembled. Label nodes as GND, 5V and A0, and designate resistors with their values.

Hint: the two resistors are in series and this is a voltage division circuit.

2.4. Theoretically, what should the voltage of the middle pin of the resistors be?

Write the voltage division formula and calculate the theoretical voltage of A0, assuming resistor values are exactly $1k\Omega$ and $4.7k\Omega$.

2.5. Read the actual voltage of A0 from Serial Monitor.

What is the actual voltage of A0? Is the actual voltage the same as the theoretical voltage? If not, what are the reasons for this discrepancy? Name two.

2.6. Assuming the value of the $4.7k\Omega$ is unknown (let's call it R_2), the actual value of the $1k\Omega$ resistor is the same as its theoretical value (R_1), and by knowing the voltage of A0 from the previous section (V_{A0}), calculate the actual value of the R_2.

Rearrange the voltage division formula so that only R_2 is on the left side of the equation and and all other terms are on the right side. Then replace the terms with their values ($R_1 = 1k\Omega$, $V_{cc} = 5V$ and V_{A0}) to obtain the actual resistance of R_2. Is this value similar to its theoretical value?

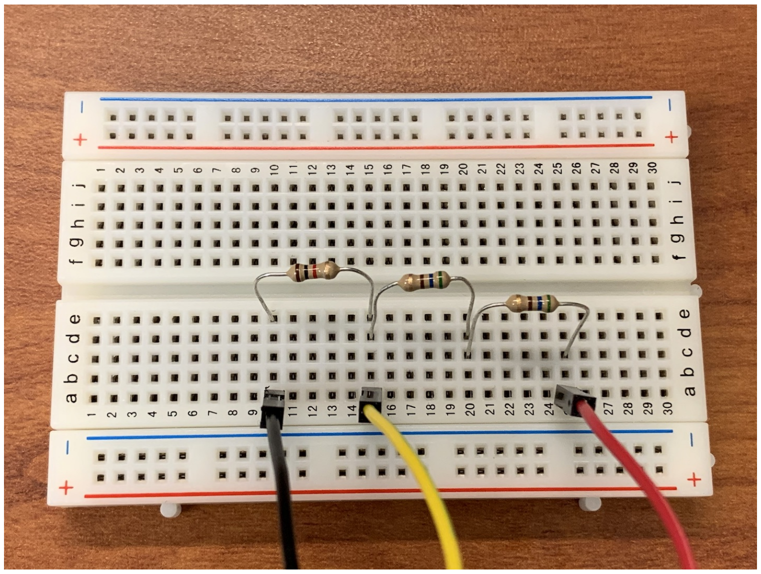
2.7. Using the resistor color table, find a 560Ω resistor from the resistor strip, replace it with the $4.7k\Omega$ resistor and repeat steps 2.4 to 2.6.

What is the theoretical and actual voltage of A0 when using a 560Ω resistor? What is the actual resistance of R_2 using the voltage division formula?

Congratulations! You have reinvented the multimeter! This technique is similar to the process that actual multimeters use for measuring resistances, although for better accuracy we would need a $1k\Omega$ resistor with very little tolerance.

3 Series Resistance

Grab another 560Ω resistor and put it in series with the previous one. On the breadboard you would have 3 resistors in series: $1k\Omega$, the first 560Ω and the second 560Ω. Connect GND to the left pin of $1k\Omega$, 5V to the right pin of the second 560Ω, and A0 to the common pin between $1k\Omega$ and the first 560Ω. Use this picture as your guide.



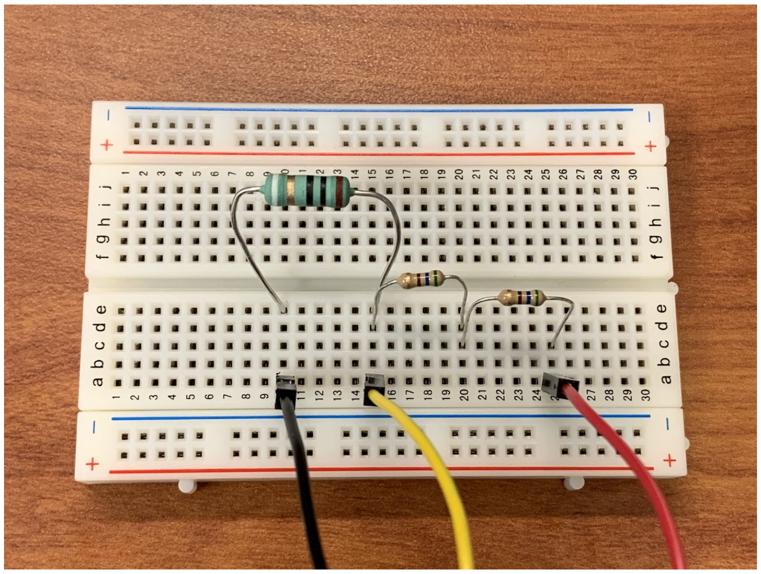
3.1. Theoretically, what would be the equivalent resistance of two 560Ω in series?

3.2. Using the voltage division formula, what is the theoretical voltage of A0?

3.3. What is the actual voltage reading of A0 from Serial Monitor?

3.4. What is the actual equivalent resistance of the two 560Ω in series? Calculate using the same technique as step 2.6.

3.5. Using our Arduino multimeter/oscilloscope, so far we've been able to measure voltages and calculate resistances. But how about currents? There is a nifty technique that's often used in electronics for converting currents to voltages, which is simply using our old friend the Ohms law! This is usually done with a relatively small resistor that minimally interferes with the circuit's function. Using the resistor color table find out what color should the bands be for a 10Ω resistor. You should be able to find one somewhere in the extra lab kit. Then, replace it with the $1k\Omega$ on your breadboard as shown in the picture.



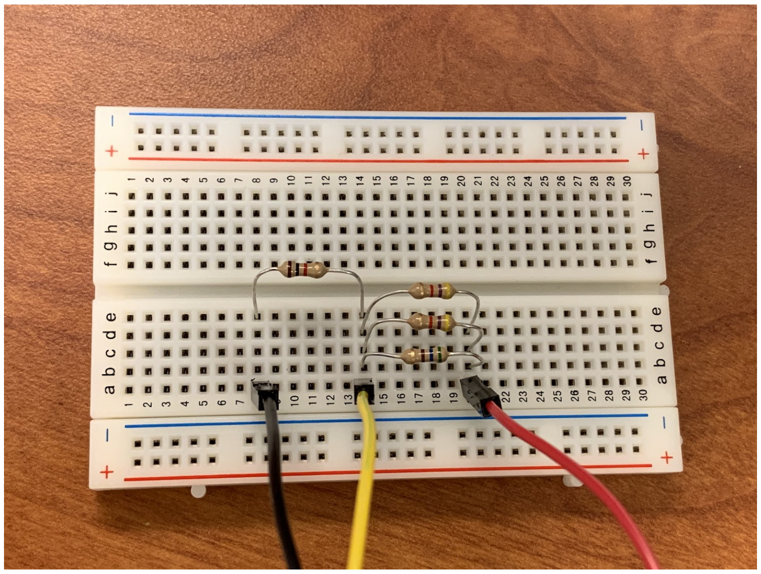
Assuming the 10Ω has minimal impact on the circuit, meaning the two 560Ω resistors are directly connected to 5V, how much current should theoretically flow through them?

3.6. By having A0 connected to the right pin of the 10Ω resistor, we are able to read the voltage across it on the Serial Monitor. This tiny voltage (probably just a few millivolts) is the result of the current that passes through the resistor, and their values are linked by the Ohms law. Since all resistors are in series, this current is the same as the current that flows through each of the two 560Ω resistors.

Using Ohms law and the voltage of A0, calculate how much current is flowing through the 10Ω resistor? Pay extra attention to the units. Compare this two your theoretical value from the previous step. If they are different, name two reasons why.

4 Parallel Resistors

In the final section of this lab we'll be verifying our knowledge of parallel resistors and their equivalent resistance. Grab two $4.7k\Omega$ and one 560Ω resistors and put them in parallel on the breadboard. Then, take a $1k\Omega$ resistor and put it in series with all the rest. Connect the two ends of your gigantic resistor network to GND and 5V, and connect the middle junction to A0. Your circuit should look like the picture below.



4.1. Draw the circuit diagram of what you have assembled on breadboard. Label GND, 5V, A0 and designate all resistors with their values.

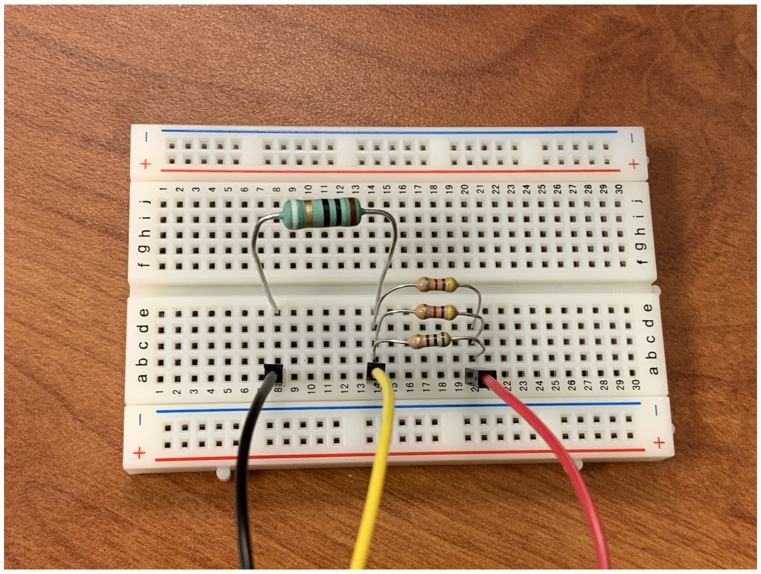
4.2. Theoretically, what would be the equivalent resistance of the two $4.7k\Omega$ and one 560Ω resistors in parallel?

4.3. Using the voltage division formula, what is the theoretical voltage of A0?

4.4. What is the actual voltage reading of A0 from Serial Monitor?

4.5. What is the actual equivalent resistance of the two $4.7k\Omega$ and one 560Ω resistors in parallel? Calculate using the same technique as step 2.6.

4.6. To measure the overall current that flows through all of our parallel resistors, replace the $1k\Omega$ with the 10Ω on your breadboard, similar to step 3.5.



Assuming the 10Ω has minimal impact on the circuit, meaning the three parallel resistors are directly connected to 5V, how much current should theoretically flow through them?

4.7. Using Ohms law and the voltage of A0, calculate how much current is flowing through the 10Ω resistor? Pay extra attention to the units.

4.8. The circuit in the next page shows how to measure the current that flows only through the 560Ω resistor. This step is optional, performing and writing the results in your report has bonus points!

BONUS: Snap a photo of what you have assembled on breadboard and draw the circuit diagram of it. Theoretically, how much current should flow through the 560Ω resistor? Based on the voltage reading from A0, how much current is actually flowing through it? Briefly describe how the circuit works.