Name/NetID:	Teammate/NetID:

Module 3C: Arduino as Voltmeter

Notes:

Laboratory Outline

As you work through the labs during the semester and some of the modules you may want to continue experimenting at home. Luckily,, the microprocessor board can be used – carefully – as test equipment making your kit a mobile laboratory. This lab will show you how to measure the voltage in the circuits used in the labs.

In addition, you will learn (or relearn) to use the Arduino Integrated Design Environment (IDE), to edit and upload programs to the processor board, the structure of the code that is interpreted by the IDE, and will be introduced to a few of the programming and debugging statements.

Notes:

The Analog Inputs on the Arduino

Included on the Arduino/RedBoard are a suite of 6 pins (lower right) labeled Analog Pins. These pins accept an analog voltage – that can be time-varying – and convert the continuous voltage into a number from 0 – 1023 every 100 microseconds. The special circuitry on the chip that can do this is called an Analog-to-Digital Converter. This is the crucial device that interfaces our analog world of touch, sound, color, heat, etc… to the digital world inside our computers. We can exploit this powerful feature to measure voltages and display them in a readable format – every 100 microseconds.

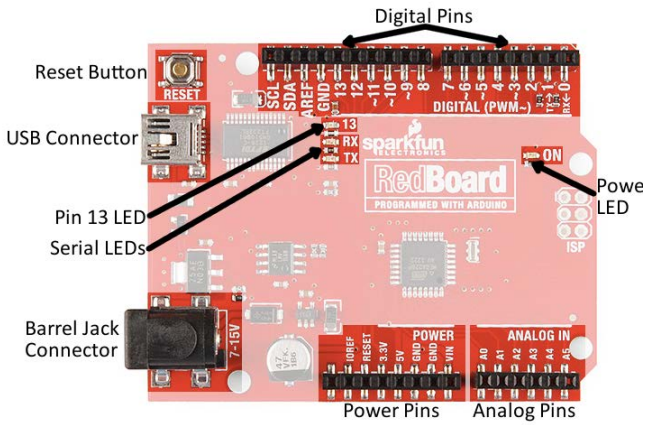


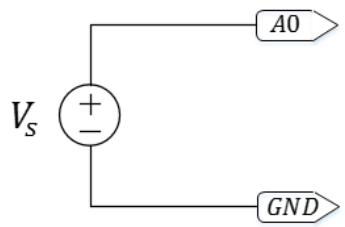
Figure 1: Physical layout of the RedBoard.

Notes:

Electrical Characteristics

Before hooking your microprocessor board between two arbitrary points in your circuits it is good to understand the limitations of the Arduino as a measuring device.

- ✓ Set the power supply to +6V mode.
- ✓ Hook-up the power supply the analog pin labelled A0 as shown in the schematic below. The arrow shapes indicate which pin on your Arduino/RedBoard to connect to the positive and negative terminals of the power supply.



- ✓ Open the Arduino IDE – look for the icon 
- ✓ A window will popup that has the barebones code needed to tell the board to do absolutely nothing. If this does not happen open the file from File > Examples > 01.Basics > BareMinimum.

Question 1: Navigate to the Arduino site – arduino.cc. Using the help section indicate purpose of the setup and loop portions of the code (yes, it was actually explained in the comments but navigate to the Arduino site to see how useful and informative it is).

Notes:

Analog to Digital conversion

This section introduces the coding needed to use the Analog Input pins, as well as illustrate how the analog voltage applied to the pins gets mapped onto a 10-bit integer.

- ✓ Below is a simple program that reads in an analog voltage and displays the 10-bit digital representation of that signal to a window provided by the IDE called the *serial monitor* in real-time. Enter this code into the stripped down code containing only an empty setup and loop sections. **NOTE:** You can dispense with the comments or write your own. The character /* indicates the beginning of an extended comment and */ indicates the end. Inline comments require to forward slashes // so the code used by the Arduino IDE to translate the program statements to machine code ignores what follows.

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

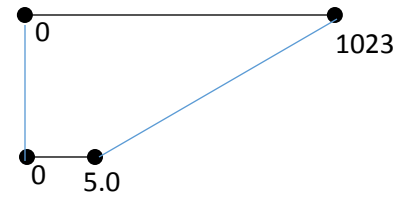
Notes:

- ✓ After checking under the Tools menu that the software knows which board you are using (the RedBoard is a clone of the Arduino Uno) and which COM port you are using. When you plug the USB cable into the lab computer the associated COM port is usually the highest numbered port. For Mac users the USB communication ports are the device file names. Upload the code to the board by clicking the  icon at the top of the window.

Question 2: The result is a bit underwhelming until you bring up the Serial Monitor by clicking on the  icon. A window pops up and should display a series of numbers. Since your power supply is set on 5V the number should not be changing. Vary the voltage down to 0V and notate the current drawn as shown by the power supply's display and the integer that shows up in the serial monitor in the table. **NEVER put more than 5V or less that 0V on the Arduino/Redboard analog input pins.**

Power Supply Voltage	Power supply Current	Integer (0 - 1023)	Voltage (0 – 5)V

Notes:



- ✓ Now add a line of code in the loop{} section just before the Serial.println statement that converts the int or integer *sensorValue* set in the code by accessing the analog input at pin A0 to a float or floating point number sensorVoltage representing the actual voltage. *You need to come up with a proportionality constant (A in the formula below) to convert the integer stored in the variable* sensorValue *to a floating point number that represents the voltage.* Mathematically, this is equivalent to mapping a number in the range (0, 1023) to a number in the range (0, 5) taking care that the Arduino knows that the result is stored as a real number (indicated by the float at the beginning of the line) not an integer.

  float sensorVoltage= A*sensorValue ;

Below is an example statement using the most common math symbols.

float myAge = (sqrt(((((3*50)+44)*200)+109))^2)-1957)/2;

- ✓ Change the Serial.println(sensorValue) to Serial.println(sensorVoltage). Set the power supply to the same voltages and finish the table.

Question 3: Does the Arduino/RedBoard have the characteristics of being a "ideal" voltmeter assuming the voltage being measured is with in the range 0-5V?

- ✓ Build a simple circuit by connecting 2 10kΩ resistors in series and the power supply set at some voltage between 0 and 5V.

Notes:

Question 4: Measure the voltage across each of the resistors and write them down. If the Arduino/Redboard configured this way acts an ideal measuring device and the resistors are identical than you would expect the voltages to be identical. Are they? If not what might account for the differences?

Question 5: Why would it be a bad idea to probe voltages in circuitry that is powered by the board you are using as a voltmeter unless you are very careful?

NOTE: unless you are very comfortable with your power and ground connections on the board and in the external circuitry DO NOT USE THIS METHOD

Safe Mode

- ✓ Disconnect the power supply from the resistor circuit and connect the 5V supplied by the board to power the circuit. The pin labeled +5V replaces the positive terminal of the power supply and any of the pins on the board labeled GND replaces the negative terminal of the power supply.

Question 6: Probe the voltages at different points in the circuit by connecting A0 to different nodes - DO NOT disconnect the GND connection. Record the values and explain why they are different from the voltages you just

measured. Hint: We can see that all of the pins – either outputs or inputs – have measurable voltages but only a single pin for each one. But to measure a voltage using a voltmeter there needs to be 2 points in the circuit to probe. So the board has a single point of reference for every voltage.

Question 7: What does the delay function do and why was it included? You can access descriptions of all of the statements at the Arduino.cc website.

Notes: