ME 120
October 16, 2019

Analog Input from a Potentiometer
Gerald Recktenwald
gerry@pdx.edu

Analog Input from a Potentiometer

A potentiometer is an adjustable voltage divider that can be used as a method of user input to an Arduino program. The left side of Figure 1 is a photograph of a typical potentiometer with pins that fit into a breadboard. The right side of Figure 1 is a schematic representation of the potentiometer.

A potentiometer has three external electrical contacts. By turning the knob, the electrical resistance between the middle contact and the two external contacts is adjusted. The change in resistance allows the potentiometer to function as a variable voltage divider as depicted by the schematic in the right side of Figure 1. The resistance between terminals A and B is fixed. In the typical potentiometer circuit, a supply (or input) voltage is applied across terminals A and B. The output voltage is measured between terminals W and B or terminals W and A.

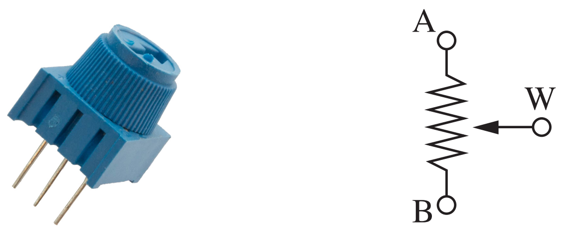

Figure 1: A typical rotary potentiometer (left). Schematic for a potentiometer (right). The middle pin is called the "wiper".

Figure 2 is a schematic of a potentiometer that can supply a variable voltage to an analog input pin of an Arduino. Terminals A is connected to a 5V supply and terminal B is connected to ground. The wiper is connected to an analog input pin. Figure 3 shows a physical representation of the wiring with the wiper of the potentiometer connected to analog input pin 3 of an Arduino.

The `potentiometer_input.ino` program in Listing 1 reads and prints the voltage between the wiper and ground for a potentiometer wired as in Figure 2 Figure 3. The reading will be an integer between 0 and 1023. To convert to voltage, multiply the reading by 5/1023.

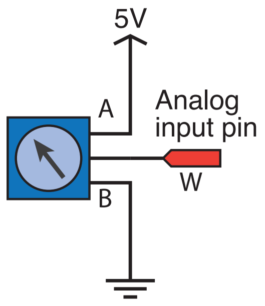

Figure 2: Schematic of a potentiometer connected to an Arduino analog input pin. The A, B, and W pins correspond to the terminals in Figure 1.

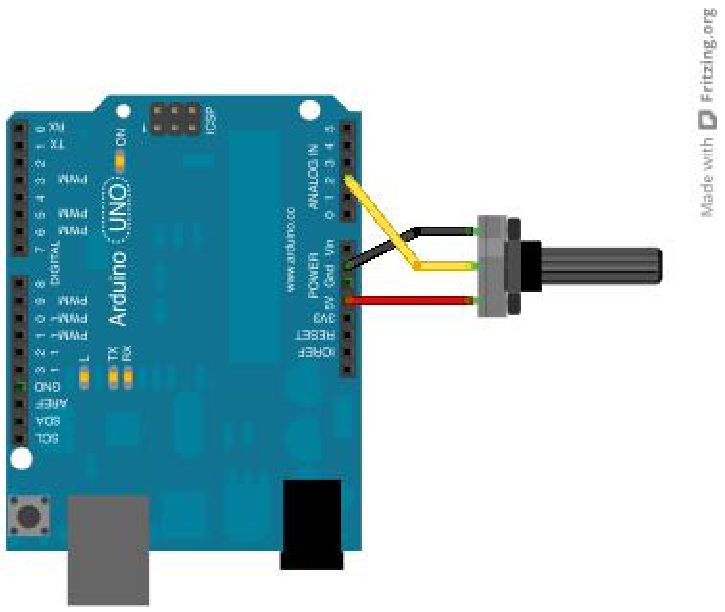

Figure 3: Arduino wiring of a potentiometer circuit for analog input on pin 3.

```
//  File:  potentiomter_input.ino
//
//  Read a potentiometer and print the raw analog input value
//  and the voltage

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int    potPin=3, potReading;
  float  potVoltage;

  potVal = analogRead( potPin );
  potVoltage = potVal*5.0/1023.0;

  Serial.print( potVal );
  Serial.print("  ");
  Serial.println(potVoltage)
}
```

Listing 1: Arduino program to read and display voltage across a potentiometer with the wiper connected to analog input pin 3.

Scaling of Arduino Analog Input Readings

The Arduino Uno can read voltages on one of six analog input pins. The maximum input voltage is 5V, and the analog input readings are returned as 10-bit integer values. The maximum numerical value for a (unsigned) 10-bit number is 1023. Therefore, when the maximum input of 5 V is applied to one of the input pins, the maximum numerical value for an analog reading is 1023. On a 10-bit scale there are 1024 numbers starting with 0 and ending with 1023.

Question: To convert a numerical value returned by the `analogRead` function to a voltage, should the scaling be 5V/1023 or 5V/1024?

Answer: The correct scaling is 5V/1023.

Consider a 4-bit analog to digital converter. An unsigned 4-bit number can have the values 0, 1, 2, ... 15 as depicted in Figure 4. Although there are 16 numbers that label points on the line, there are only 15 equal-length segments. Therefore, the scale factor for a 4-bit analog-to-digital converter is 5V/15 because the 5V range is divided into 15 intervals. By analogy, the correct scaling for a 10-bit analog to digital converter having a maximum input range of 5V is 5V/1023.

There are 16 values on the 4-bit number line, but the first value, 0, just marks the beginning of the line. The other values, 1, 2, 3, ... 15 are labels for the right hand endpoints of the intervals.

Yet another way to think about this is to consider the question: How many years have you lived when you reach your nth birthday? For example, on your twenty-first birthday, you have lived 21 years. The day you were born is birthday 0.

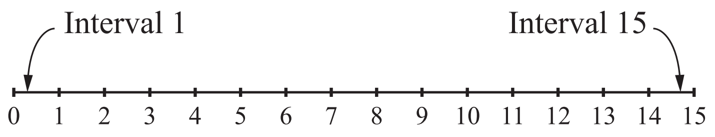

Figure 4: A 4-bit number line.

Exercises

1. In many example Arduino programs, analog readings are made with statements like `x = analogRead(A0)`, `x = analogRead(A1)`, `x = analogRead(A2)`, ... What are `A0`, `A1`, `A2`, etc.? Should users define `A0`, `A1`, `A2`, etc. as global variables?

2. In the example program in Listing 1, why are `potPin`, `potReading` and `potVoltage` not defined as global variables?

3. The maximum safe voltage for an input signal on any analog input pin is 5V. How does the design of the circuit in Figure 3 guarantee that the 5V limit will not be exceeded?

   Note: Many new microcontrollers are limited to 3.3V.

4. Some microcontrollers have 12-bit analog input resolution. What is the appropriate scaling between the analog reading and voltage for a 12-bit analog input?