

Arduino 101

Hands-on: LED Level from Potentiometer

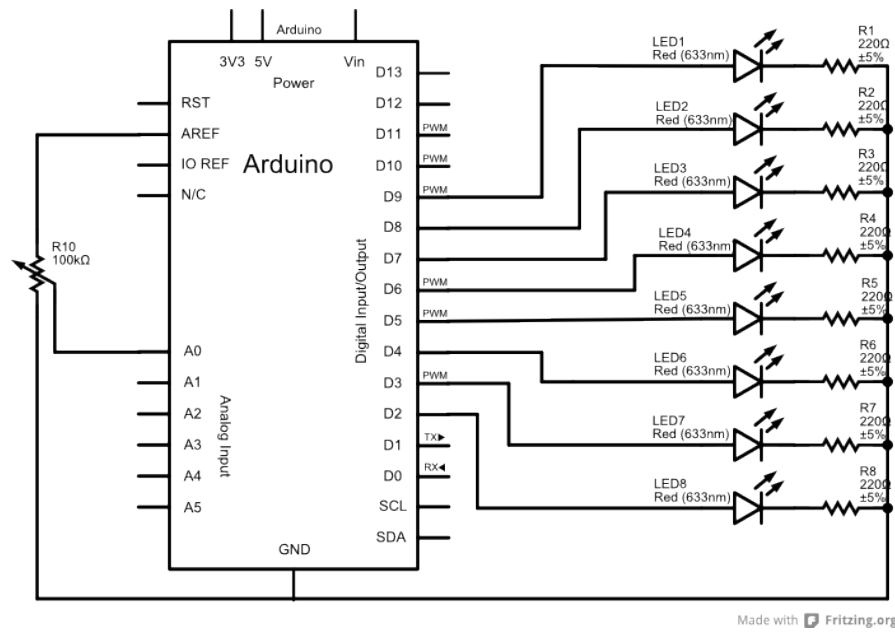
Project Description

In this project we're going to use the LED Level meter we built in the previous project to actually measure something. In this case, we're going to measure the level indicated by a 10kΩ linear potentiometer.

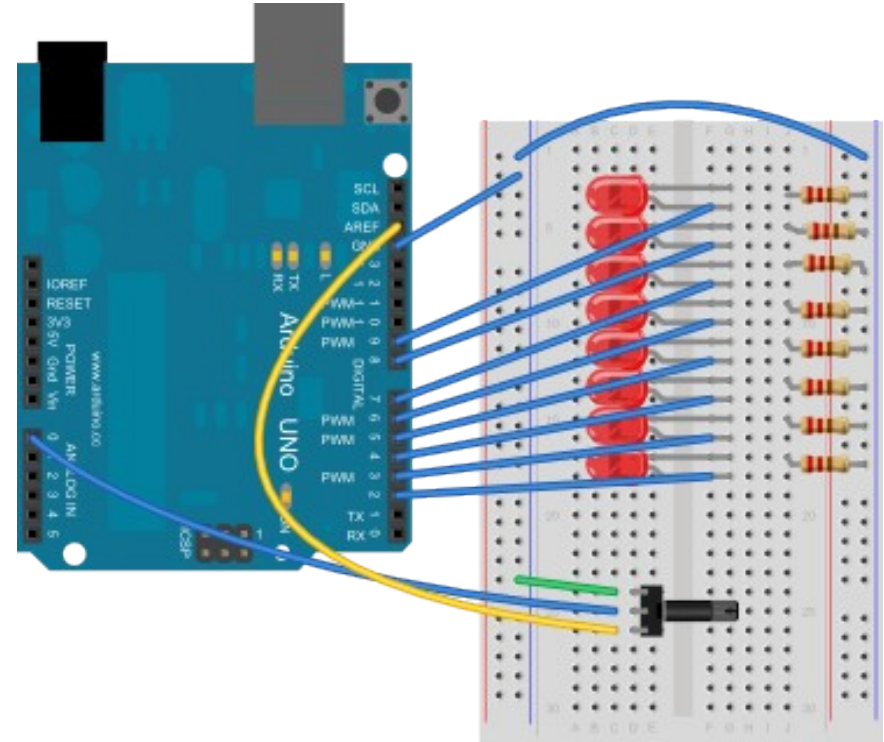
Required Parts

- 8 red LEDs
- 8 220Ω Resistors (red, red, black)
- 1 10kΩ linear taper potentiometer

Schematic



Circuit



NOTES: The LED circuit is the same as the previous project, but this time we're using a potentiometer. The potentiometer in your kit has mounting brackets attached, but they can be easily removed by bending them back and forth gently until they break off. The outer pins will line up with the holes on the breadboard, but the middle pin is between holes. Insert the potentiometer gently into the breadboard and the third pin will bend slightly to the left or right; it doesn't matter which direction, as long as you connect it to the correct wire connecting to A0 on the Arduino.

Code

```
// LED Level from Potentiometer Sketch for Arduino 101
// by Nick Borko

// function to set the LEDs to a specific level
void setLed(int level) {
  // iterate through the pins
  for (int pin = 2; pin <= 9; pin += 1) {
    // compare the pin to the level...
    if (pin < level + 2) {
      // LED is on
      digitalWrite(pin, HIGH);
    } else {
      // LED is off
      digitalWrite(pin, LOW);
    }
  }
}

void setup() {
  int pin;
  // initialize pins 2-9 to be output pins
  for (pin = 2; pin <= 9; pin += 1) {
    pinMode(pin, OUTPUT);
  }
}

void loop() {
  // read the value from the potentiometer
  // this will be a value from 0-1023
  int raw = analogRead(A0);
  // we add one to the reading to make it an even 1-1024,
  // then divide this value by 128 since we have 8 LEDs
  int scaled = (raw + 1) / 128;
  // set the LED Level
  setLed(scaled);
  // smooth out the reading a bit
  delay(50);
}
```

Discussion

A potentiometer is a variable resistor. When you turn the shaft of the potentiometer you are changing where the resistive material is being measured by the middle connection, and thus changing the resistance and output voltage. There are 2 types of potentiometers, **linear** and **logarithmic**. You are using a linear potentiometer, which changes the resistance proportional to the turn of the shaft. Logarithmic potentiometers are generally used in audio applications, such as a volume switch.

A potentiometer actually works as 2 resistors, one on each side of the middle terminal. Depending on which of the other 2 terminals you place the input voltage, you can change how the direction of the turning of the shaft affects the resistance. In this project we have it wired so that a clockwise turn *decreases* the measured resistance.

The Arduino is capable of reading analog values based on a reference voltage, which is supplied by the **AREF** connection. The value of the **AREF** voltage is read as the value 1023, which is the maximum value that can be read. When the analog input is **grounded**, or reads 0 voltage, the value is 0. Anywhere in between is a value between 0 and 1023. This is read by calling the **analogRead()** function on pin **A0**. Note that we didn't (and you should not) set the **pinMode()** for reading analog pins.

When the potentiometer is turned all the way in the direction where there is no resistance, the Arduino will read a value of 1023. When it is turned all the way in the other direction where there is the most resistance, no voltage comes through and the Arduino will read a value of 0. The value of the resistor is not important, but using a large value (such as 10k Ω) will minimize the total power consumed in the resistor.

You now have a meter capable of showing the relative value being read on pin **A0** of the Arduino. Swap out the bottom 4 LEDs with green LEDs, and the next 2 with yellow LEDs. Now it's starting to look like something useful!

Copyright ©2012 by Nicholas Borko. All Rights Reserved.

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>