LED Bar Graph

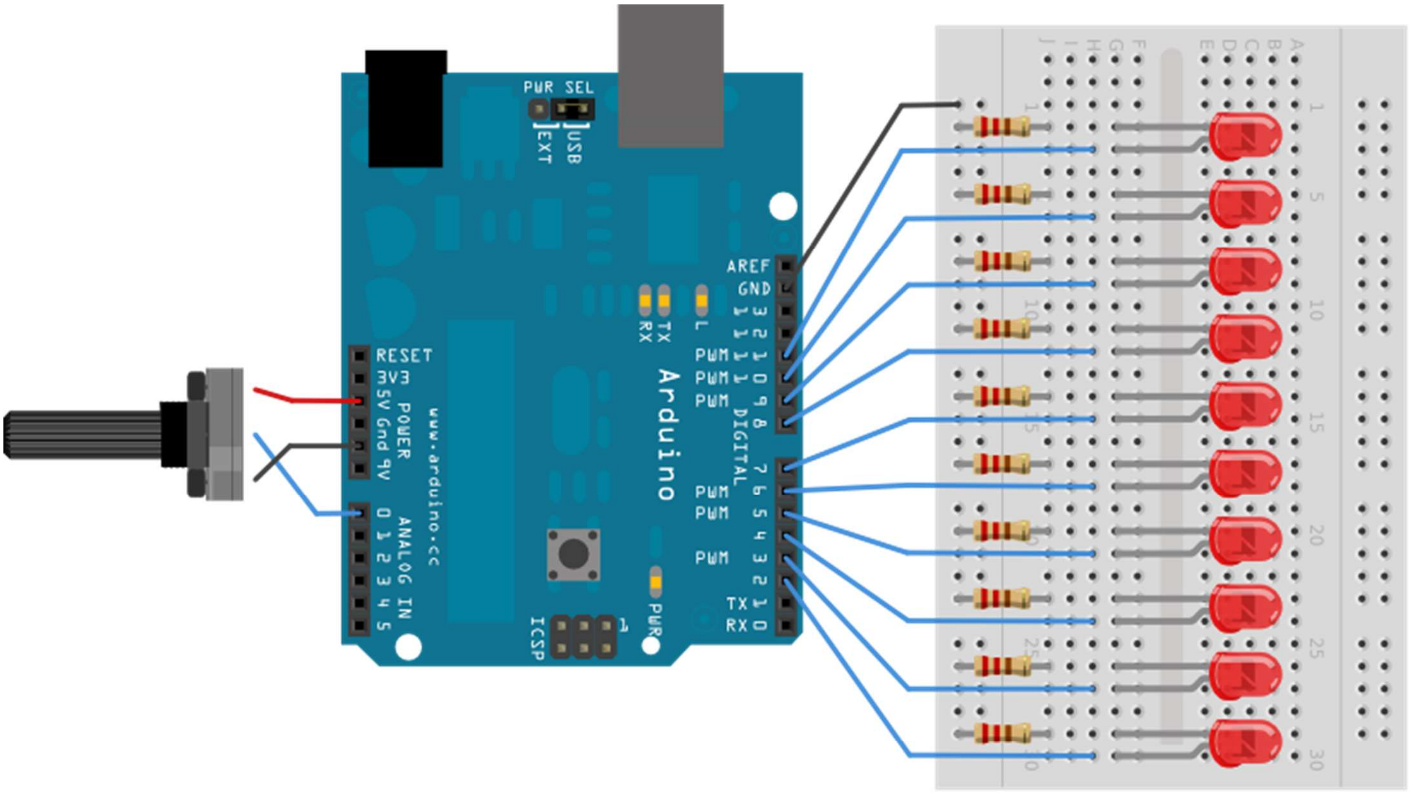
The bar graph - a series of LEDs in a line, such as you see on an audio display - is a common hardware display for analogue sensors. It's made up of a series of LEDs in a row, an analogue input like a potentiometer, and a little code in between. This tutorial demonstrates how to control a series of LEDs in a row, but can be applied to any series of digital outputs.

Hardware Required

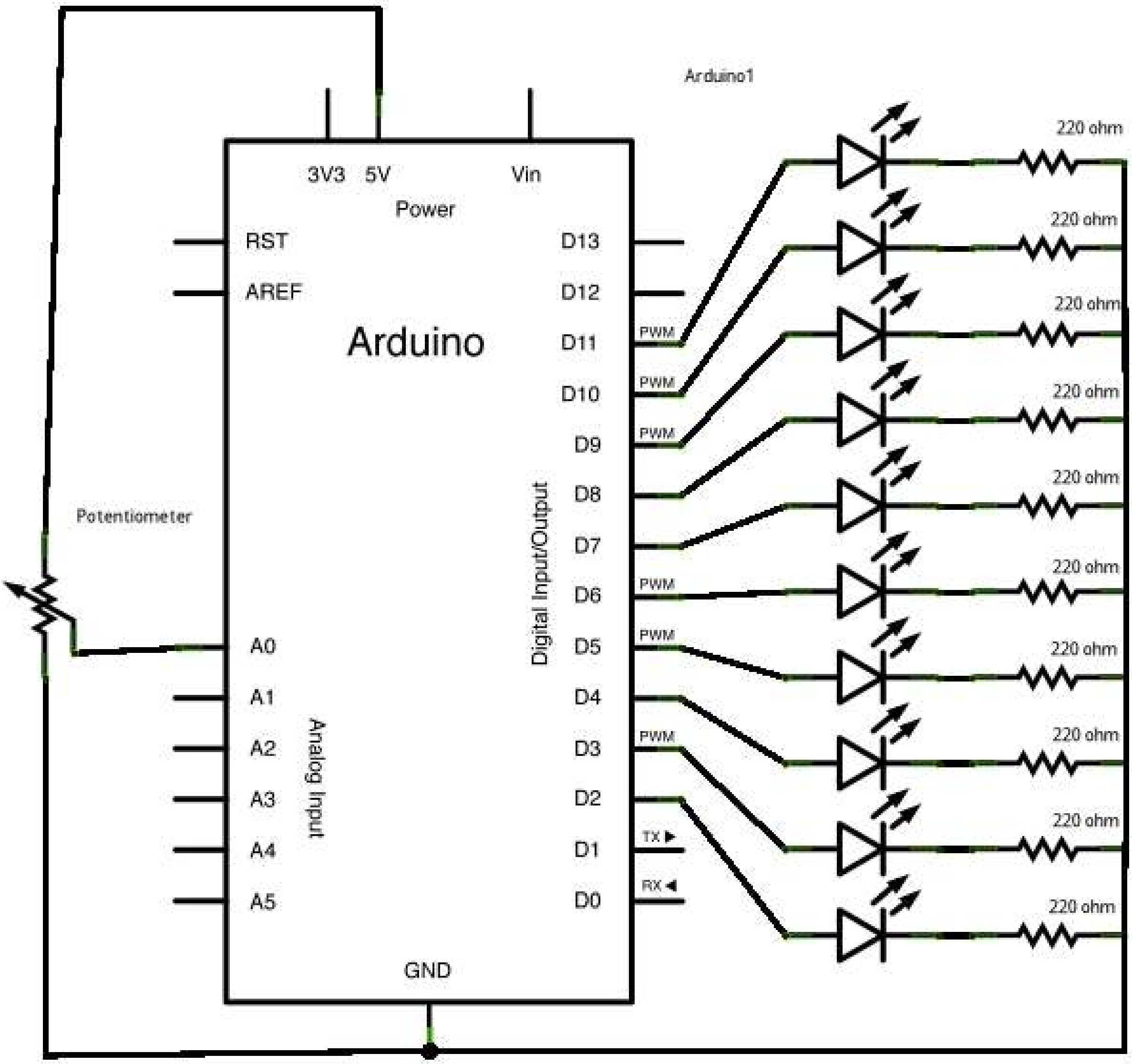
1) Arduino Board
2) LED bar graph display or 10 LEDs
3) Potentiometer
4) 10 220 ohm resistors
5) hook-up wires
6) breadboard

Circuit

click the image to enlarge


PWR SEL
EXT
USB
AREF
GND
13
12
11
10
9
8
PWM
PWM
PWM
DIGITAL
7
6
5
4
3
2
1
0
PWM
PWM
PWM
TX
RX
L
TX
RX
Arduino
RESET
3V3
5V
Gnd
9V
POWER
www.arduino.cc
ANALOG IN
0
1
2
3
4
5
ICSP
PWR
1
A
B
C
D
E
F
G
H
I
J
1
5
10
15
20
25
30

Schematic:


Arduino1
Arduino
Power
3V3
5V
Vin
RST
AREF
A0
A1
A2
A3
A4
A5
Analog Input
GND
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
PWM
TX ▶
RX ◀
Digital Input/Output
Potentiometer
220 ohm

Code

The sketch works like this: first you read the input. You map the input value to the output range, in this case ten LEDs. Then you set up a for loop to iterate over the outputs. If the output's number in the series is lower than the mapped input range, you turn it on. If not, you turn it off.

```
/*
  LED bar graph

  Turns on a series of LEDs based on the value of an analog sensor.
  This is a simple way to make a bar graph display. Though this graph
  uses 10 LEDs, you can use any number by changing the LED count
  and the pins in the array.
  This method can be used to control any series of digital outputs that
  depends on an analog input.
  The circuit:
   * LEDs from pins 2 through 11 to ground
 */

// these constants won't change:
const int analogPin = A0;   // the pin that the potentiometer is attached to
const int ledCount = 10;    // the number of LEDs in the bar graph

int ledPins[] = {
  2, 3, 4, 5, 6, 7, 8, 9, 10, 11
};   // an array of pin numbers to which LEDs are attached


void setup() {
  // loop over the pin array and set them all to output:
  for (int thisLed = 0; thisLed < ledCount; thisLed++) {
    pinMode(ledPins[thisLed], OUTPUT);
  }
}

void loop() {
  // read the potentiometer:
  int sensorReading = analogRead(analogPin);
  // map the result to a range from 0 to the number of LEDs:
  int ledLevel = map(sensorReading, 0, 1023, 0, ledCount);

  // loop over the LED array:
  for (int thisLed = 0; thisLed < ledCount; thisLed++) {
    // if the array element's index is less than ledLevel,
    // turn the pin for this element on:
    if (thisLed < ledLevel) {
      digitalWrite(ledPins[thisLed], HIGH);
    }
    // turn off all pins higher than the ledLevel:
    else {
      digitalWrite(ledPins[thisLed], LOW);
    }
  }
}
```