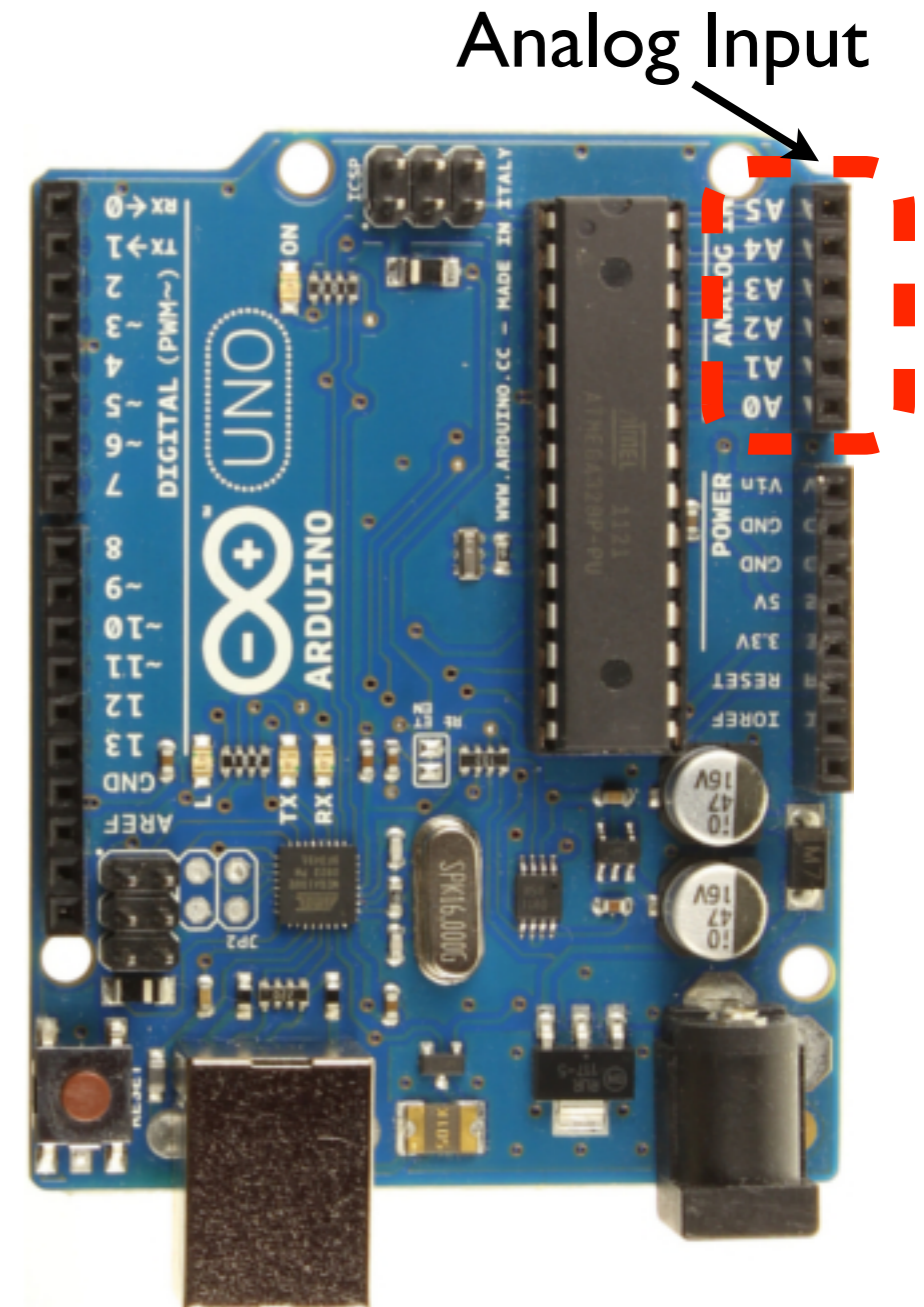


Analog

Day #2

Analog input

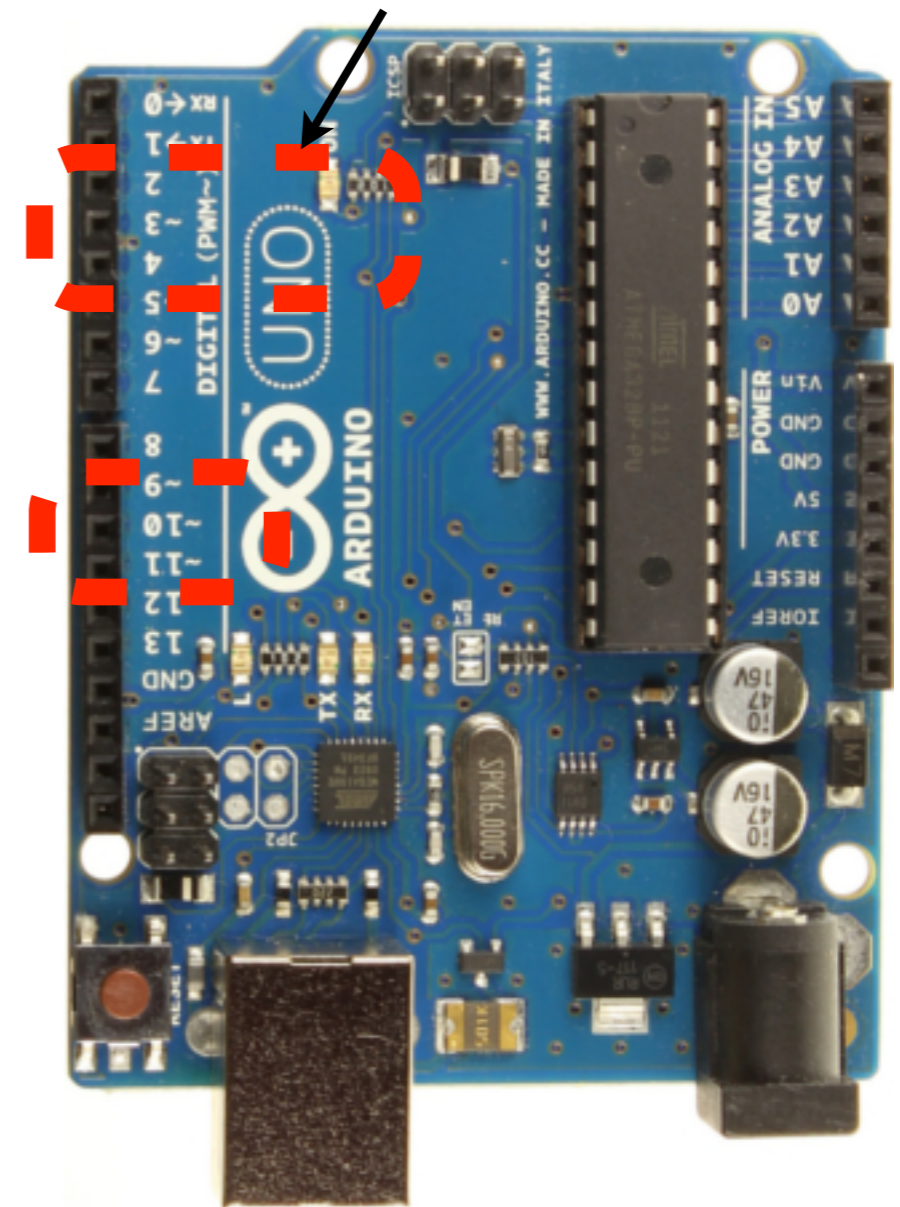
- You've seen how Arduino can output HIGH/LOW signals... what about a medium signal?
- Easy... with analog I/O!
- Output : 0 to 255
- Input: 0 to 1023
- Analog **input**: we can use it to determine the distance of an object via infra-red sensor.
- Analog **output**: we can use it to set speed of a motor or the brightness of a LED.



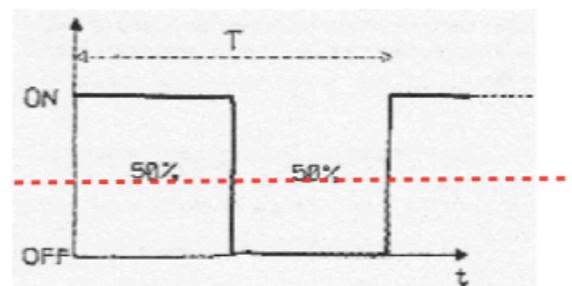
Analog output

- Some digital ports can be programmed to output analog signals.
- We enable those ports for output with pulse width modulation (PWM).
- PWM is obtained by varying between HIGH and LOW at the appropriate interval of time.

Digital pins with PWM

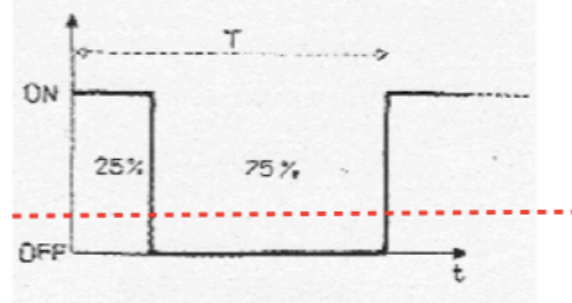


50% Duty Cycle



Effective Voltage

25% Duty Cycle



Effective Voltage



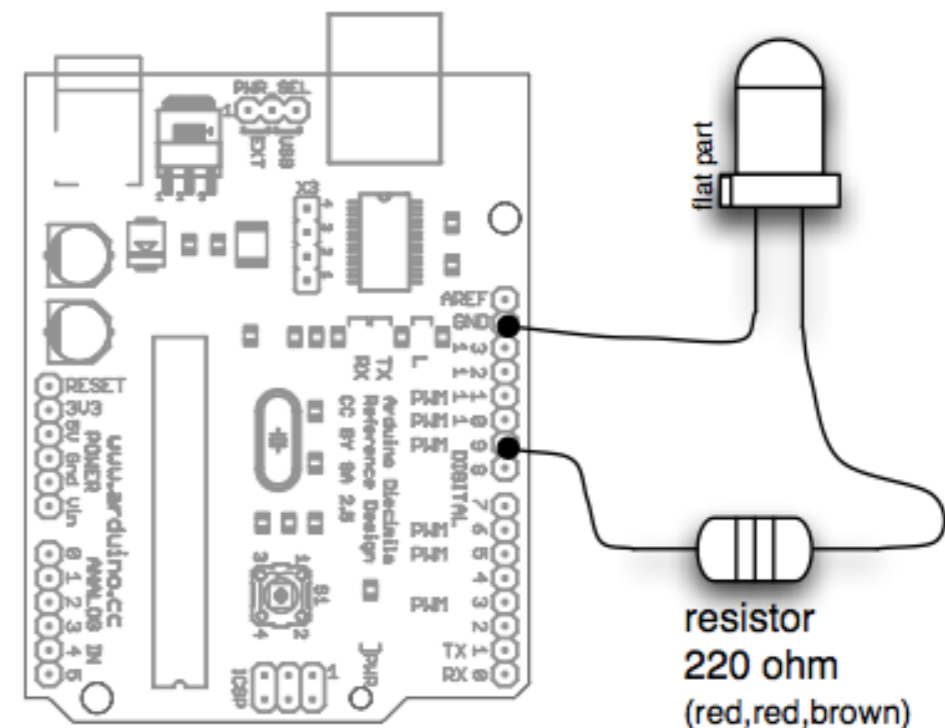
Dimming light

```
int ledPin=9;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  int value=0;

  //loops value from 0 to 255
  for (value=0; value<255;value++)
  {
    //write the index of the loop to the pin
    analogWrite(ledPin,value);
    //wait 10ms
    delay(10);
  }
}
```

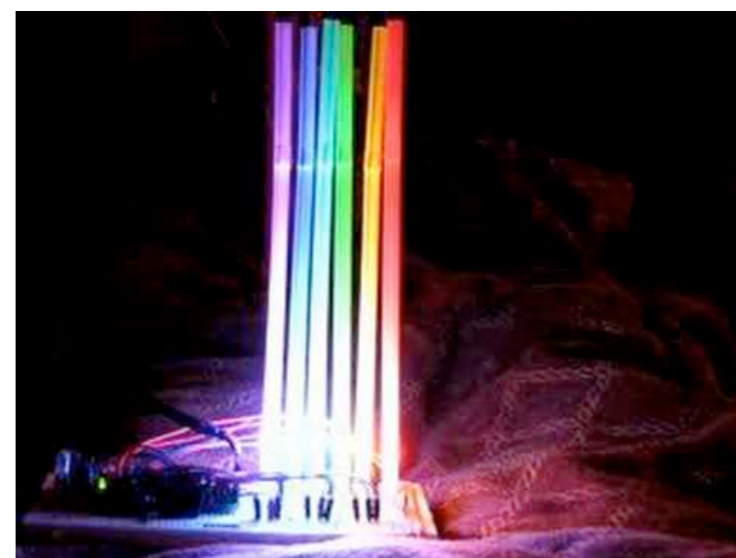
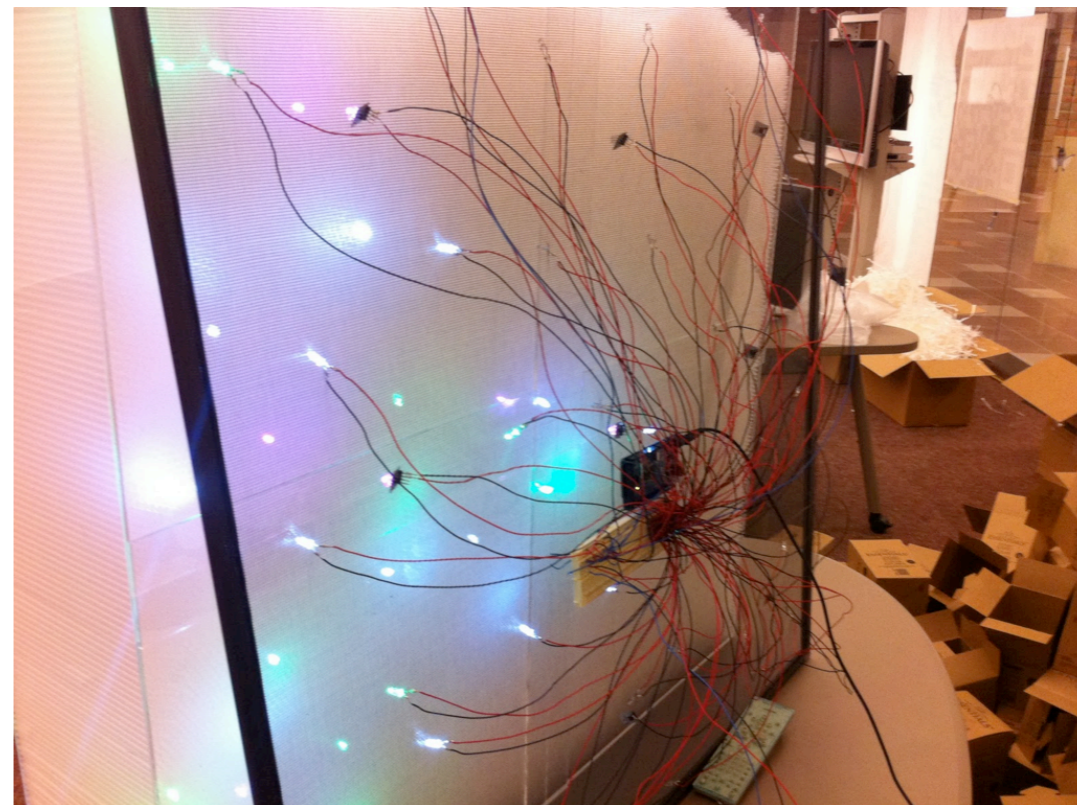


wiring diagram

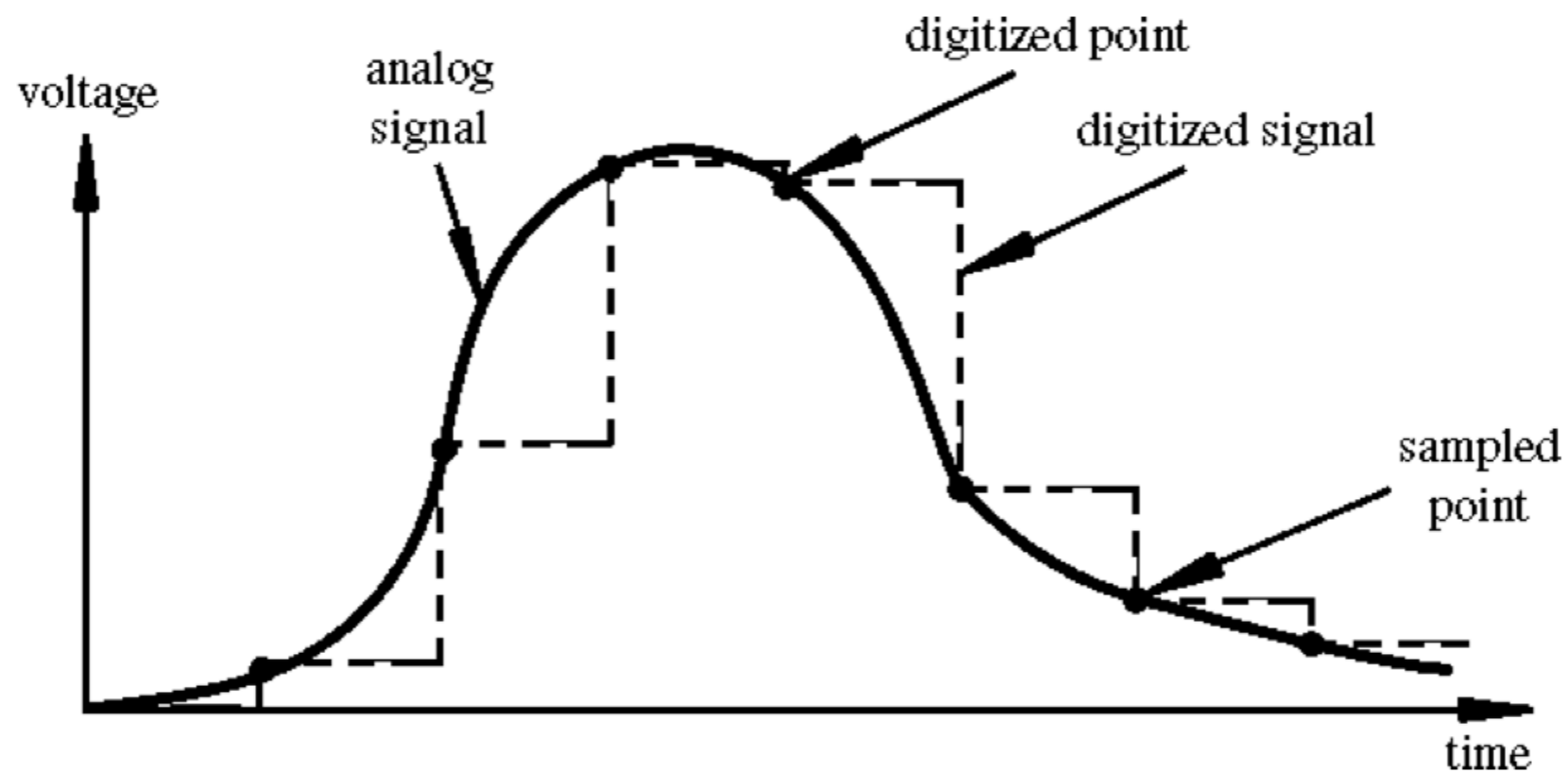


Dimming light

- Wire 2 LEDs at different PWM pins.
- Specify different delays to emulate candlelight flicker.
- Measure Voltage on a PWM pin. What's going on?
- You could make something cool like the picture and decorate your wall!
- Put straws on the LEDs for a diffused look



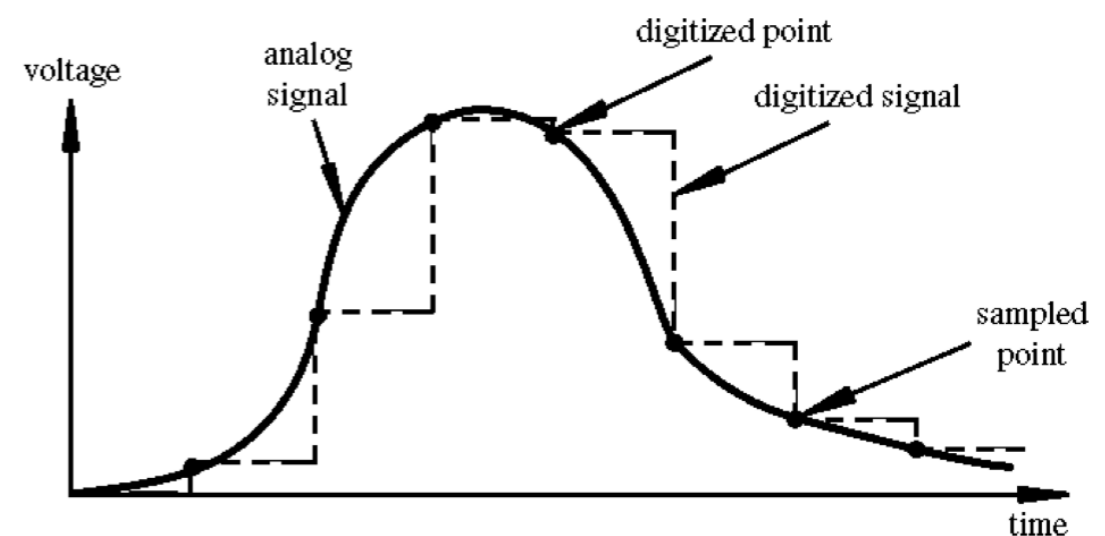
Analog input



Arduino splits analog signals into digital points

Analog input

- Many states, not just two (HIGH/LOW)
- Number of states (or values, or “bins”) is resolution
- Common computer resolutions:
 - 8-bit = $2^8 = 256$ values
 - 16-bit = $2^{16} = 65,536$ values
 - 32-bit = $2^{32} = 4,294,967,296$ values

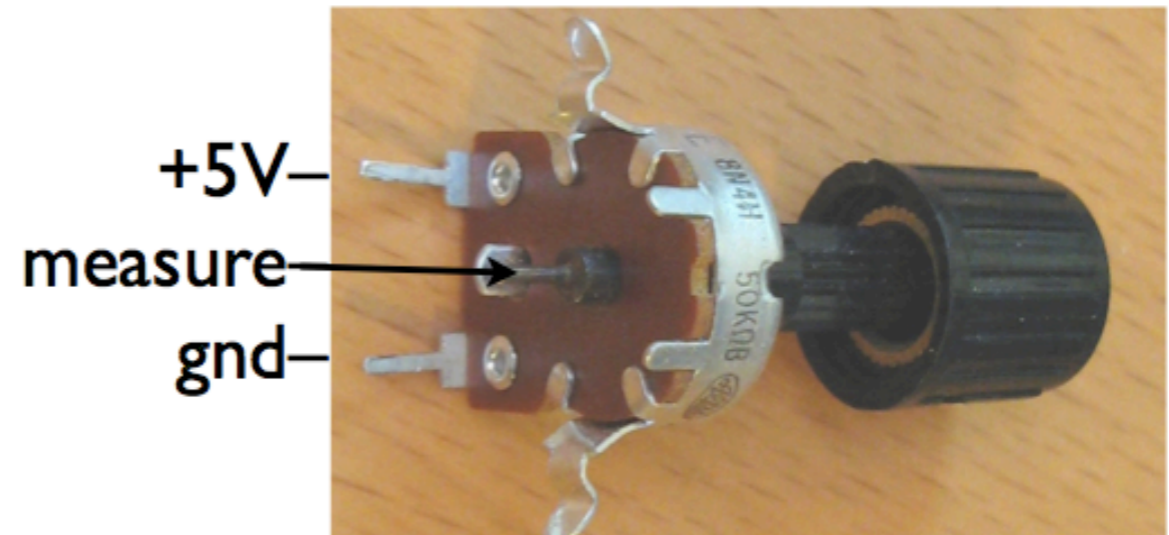
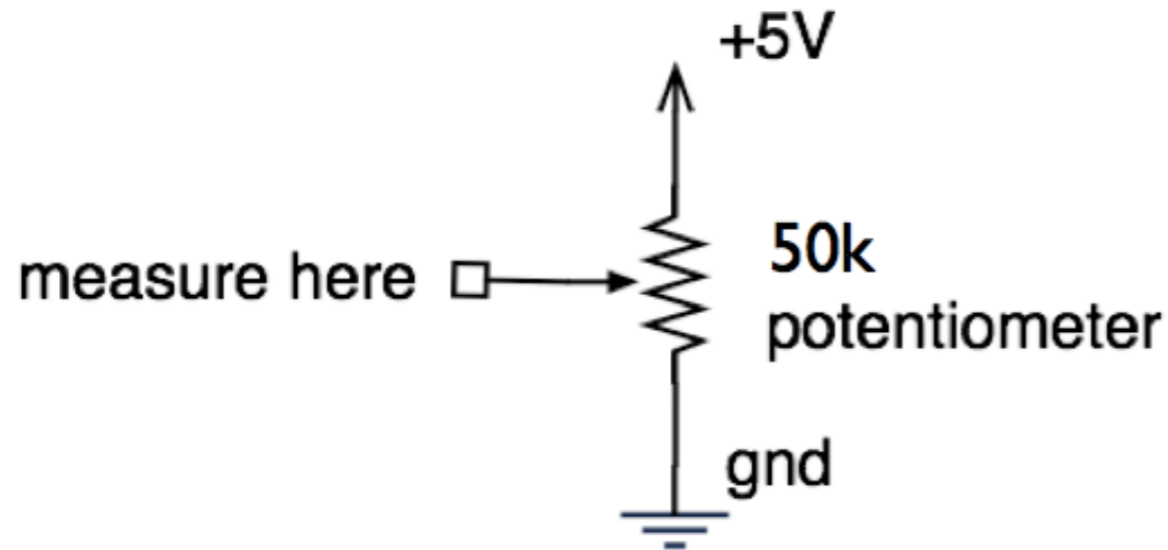


Analog input

- Arduino has six ADC inputs (ADC = Analog to Digital Converter)
- Reads voltage between 0 to 5 volts
- Resolution is 10-bit (1024 values)
- In other words, $5/1024 = 4.8$ mV smallest voltage change you can measure
- Examples of varying voltage inputs: Temperature sensors, light sensors, ...

Potentiometers

With a potentiometer we can create an analog input.



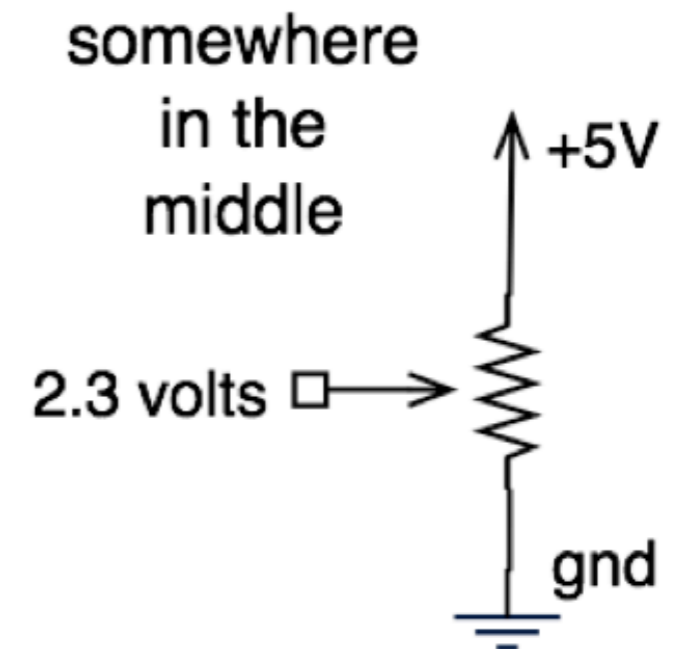
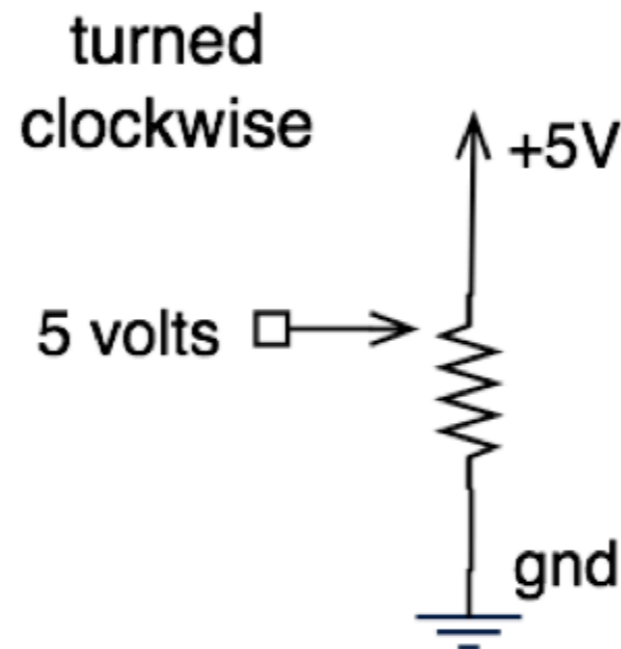
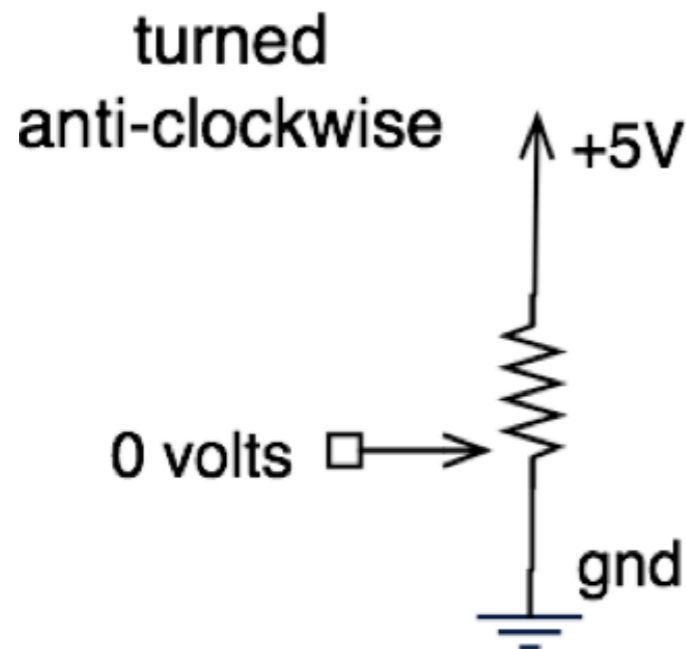
The pot you have



pots also look like this

Potentiometers

Moving the knob is like moving where the arrow taps the voltage on the resistor



Analog read code

```
int ledPinA = 9; //output pin (led)
int sensorPinA = 2; // input pin for the potentiometer
int sensorValueA = 0; //analog valueA

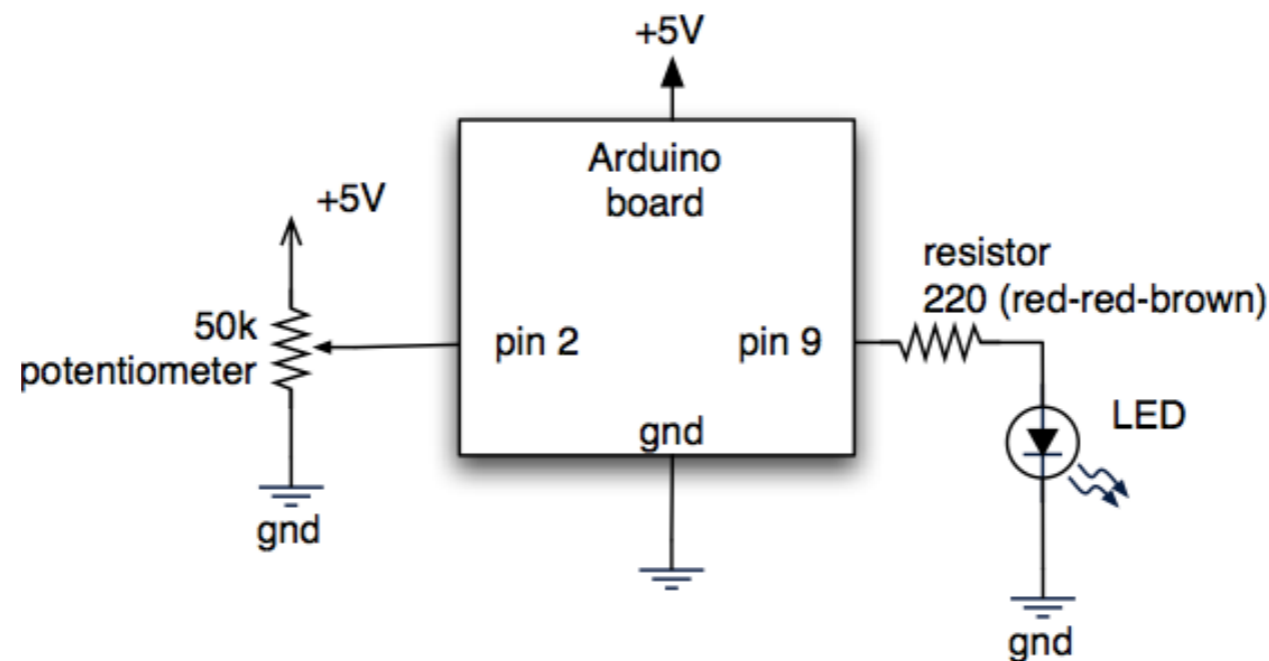
void setup() {
  Serial.begin(9600);
  pinMode(ledPinA, OUTPUT);
}

// the loop() method runs over and over again,
void loop()
{
  sensorValueA = analogRead(sensorPinA);
  digitalWrite(ledPinA, HIGH);
  Serial.println(sensorValueA);
  delay(sensorValueA);
  digitalWrite(ledPinA, LOW);
  delay(sensorValueA);
}
```



Potentiometer project

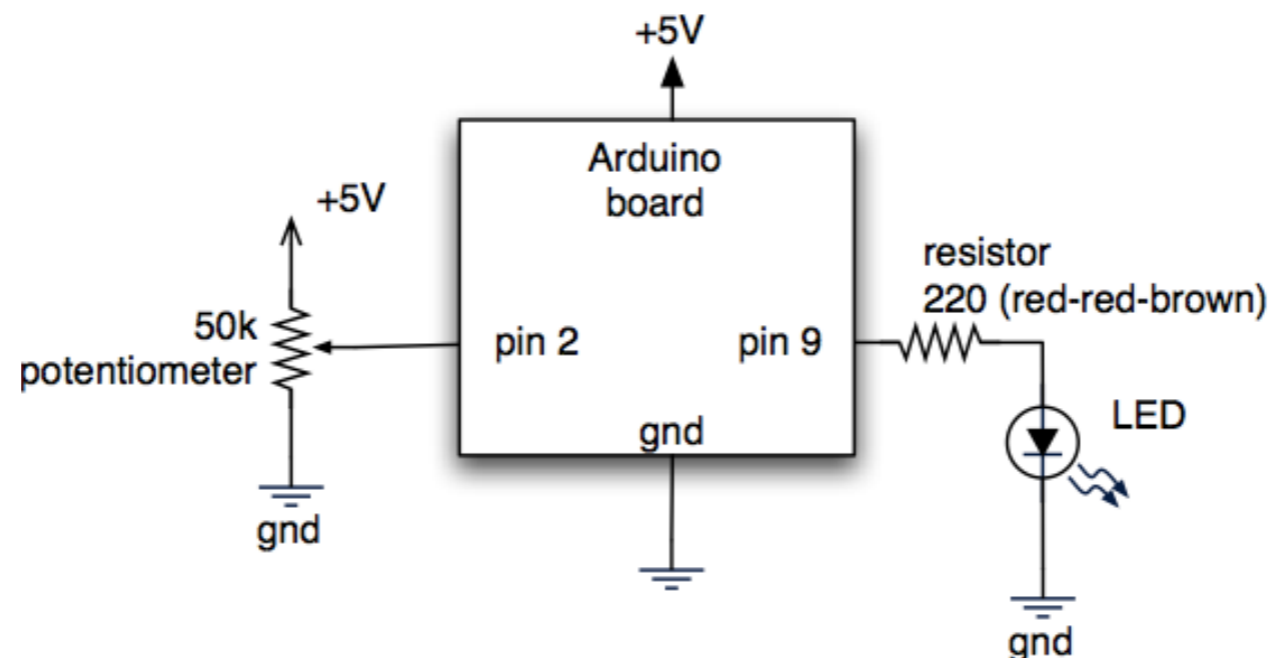
- Prototype the circuit.
- Program the following:
 - As you rotate the potentiometer, you want the LED to blink faster.
 - Display the analog input value in the serial connection.



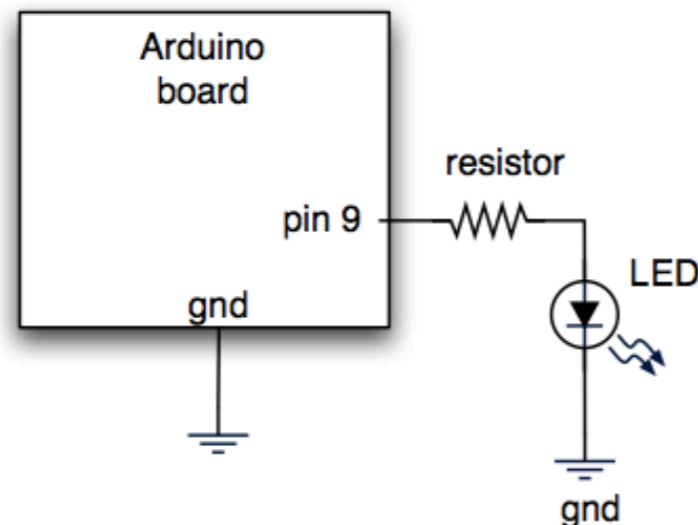


Expanding the project

- Add one more LED in a different pin and program the following:
 - If analog input is smaller than 100, turn ON one LED.
 - If analog input greater than 100 turn ON the other LED.
- Replace one LED with a speaker. What happens?



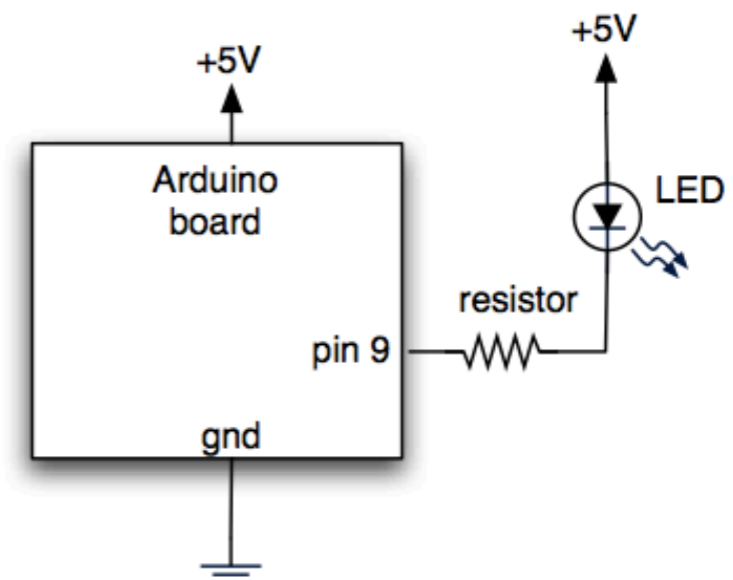
Hooking up LEDs



To turn ON: `digitalWrite(9,HIGH)`

To turn OFF: `digitalWrite(9,LOW)`

To set brightness: `analogWrite(9,val)`



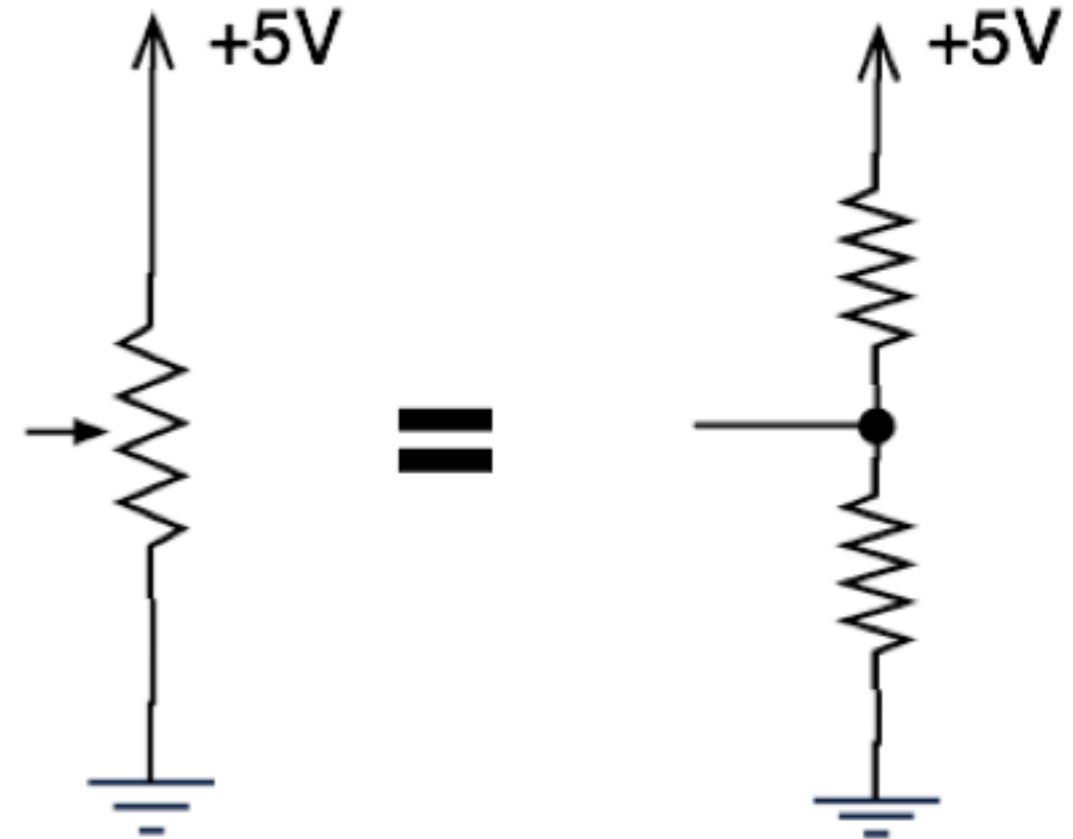
To turn ON: `digitalWrite(9,LOW)`

To turn OFF: `digitalWrite(9,HIGH)`

To set brightness: `analogWrite(9,255-val)`

Voltage divider

- Potentiometers are example of a voltage divider.
- Voltage divider splits a voltage in two.
- Same as two resistors, but you can vary them...



Sensing the dark: photocells

- Photoresistor, light-dependent resistor
- A variable resistor
- Brighter light == lower resistance
- Photocells you have range approx. 0-10k-1M



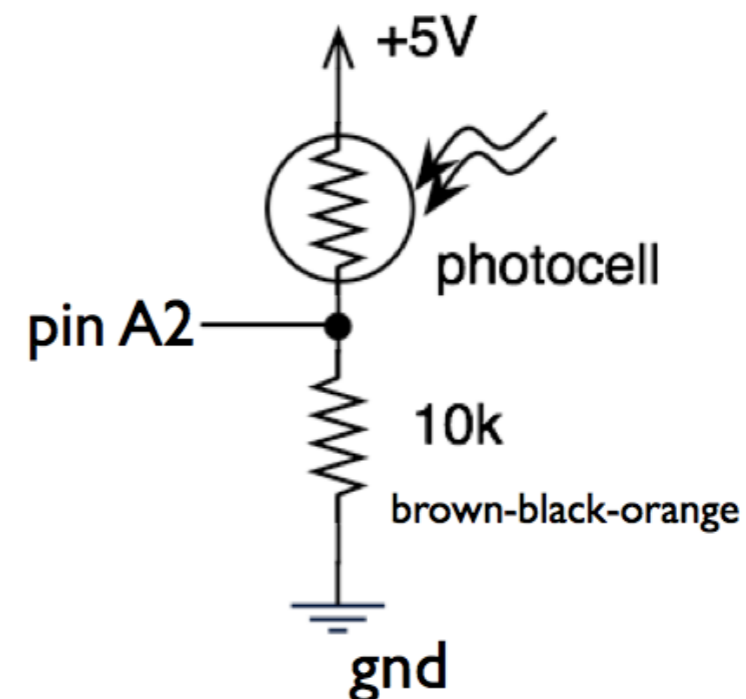
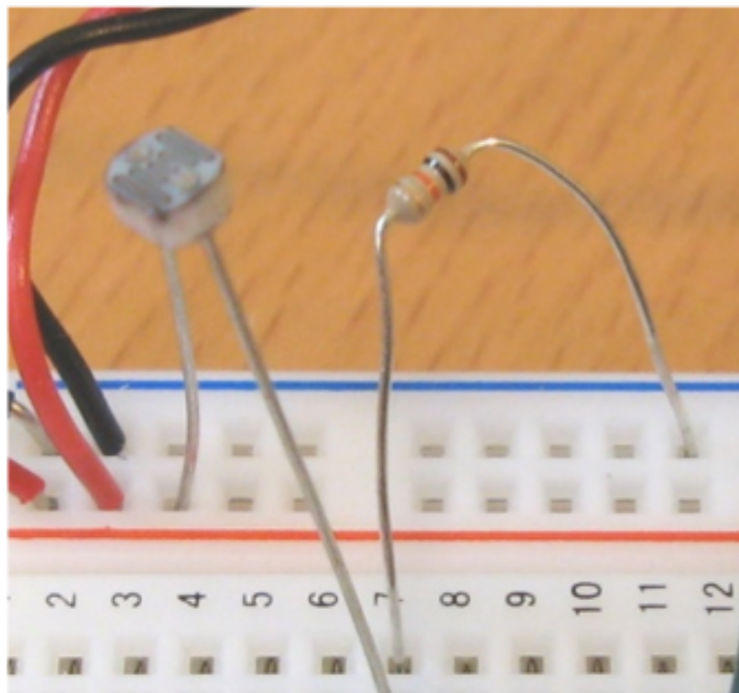
schematic symbol



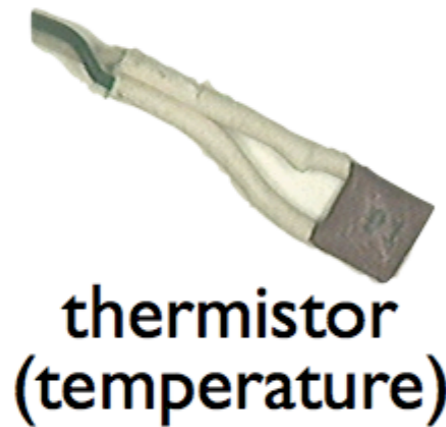


PhotoCell circuit

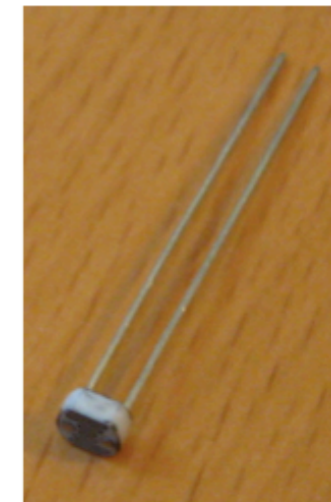
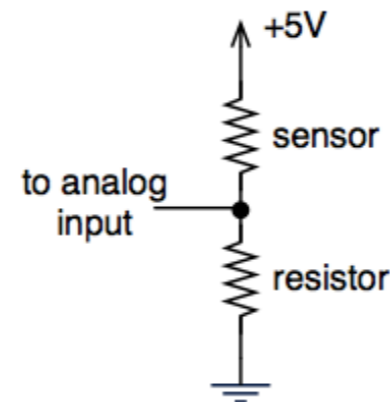
- Implementation is very similar to a potentiometer
- Pin A2 is the input to an analog port.
- See what analog voltage ranges you can get using the serial monitor.



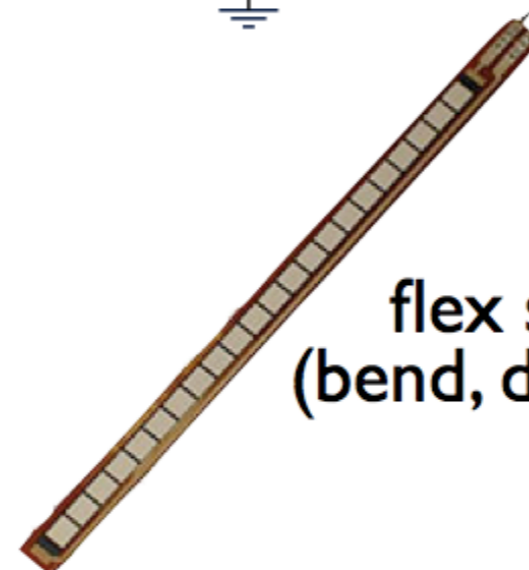
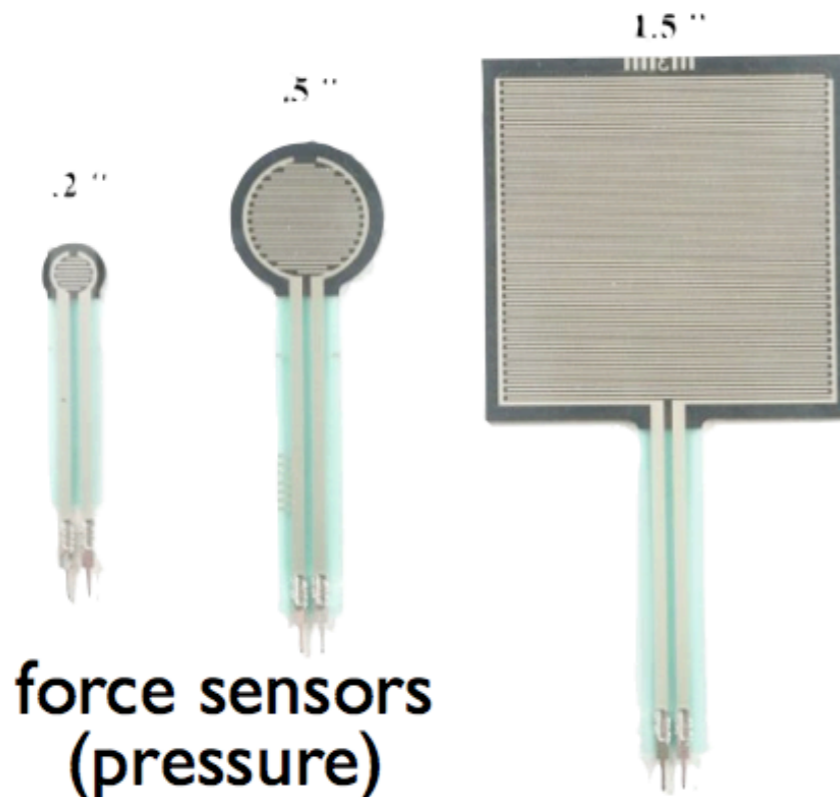
Resistive sensors



circuit is the same
for all these



photocell
(light)



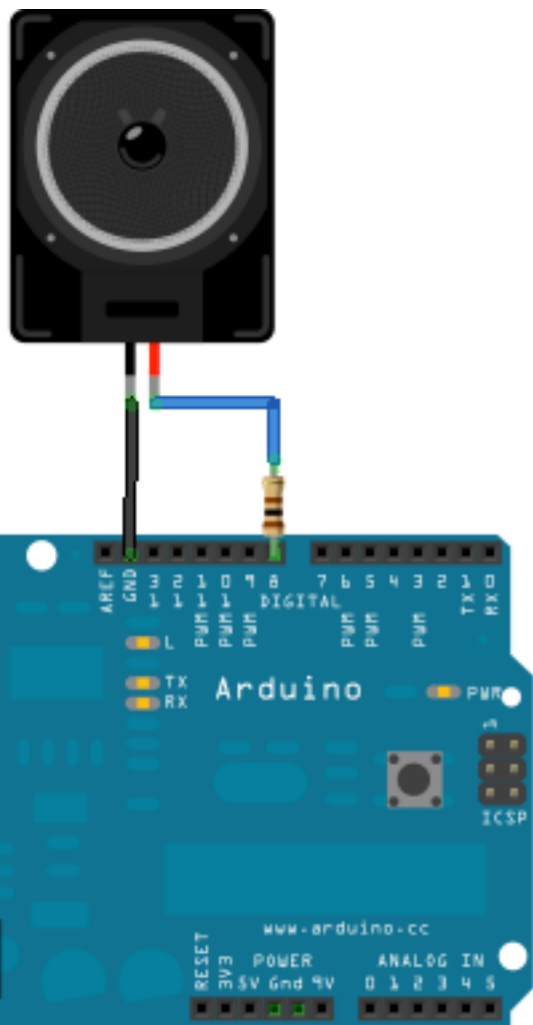
flex sensor
(bend, deflection)

also air pressure
and others



Lets play some music

Download pitches.h from: <http://arduino.cc/en/Tutorial/Tone>



```
#include "pitches.h"
int melody[]=
{NOTE_C4,NOTE_G3,NOTE_G3,NOTE_A3, NOTE_G3};

//durations: 4 = quarter note, 8 = eighth note
int noteDurations[] = {4,8,8,4,4,4,4,4 };

void setup() {
  for (int thisNote = 0; thisNote < 8; thisNote++)
  {
    int noteDuration = 1000noteDurations[thisNote];
    tone(8, melody[thisNote],noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(8);
  }
}

void loop() {}
```

Theremin

- “ooo-weeee-oooooo”
- The original spooky sound machine
- Works by measuring your body’s electric field
- We’ll use a resistive sensor in lieu of RF
- Leo Theremin in his own words - [video](#)



Leon Theremin



Play a tune

- Read an analog signal generated through a resistive sensor
- Blink an Arduino PIN very fast relative to the analog signal
- Put a LED in that pin.
- Replace the LED with a speaker.
- Play your favorite tune!

