Practical Electronics & Programming

with Arduino

Session 1: Hello World

The Instructor

Joseph (Joe) Schlesinger

- B.S. Electrical and Computer Engineering Worcester Polytechnic Institute
- Former professional Firmware Engineer
- Developing Arduino-compatible robotics board
- Former electronics for non-majors TA

Class Expectations

What this class will get you:

- Build your own simple Arduino projects
- Focused on inputs->outputs
- Basic electronics concepts
- Basic programming concepts

What it won't get you:

- Advanced circuit analysis
- Hardcore programmer kung-fu
- Get you an electronics/programming job

Class Overview

May 13th

Session 1: Hello World

Setup your Arduino, and do basic programing, presented controlling LEDs and a buzzer

May 27th

Session 2: Communicate

Talk to your computer with Serial, and talk to a person with a display

June 3rd

Session 3: Yes/No

Control projects using buttons, switches and distance

Output using lights and buzzer

Class Overview (continued)

June 10th

Session 4: 0%-100%

Control projects using temperature, magnets, light and knobs

June 17th

Session 5: *Whirrrrrrr*

Motor and servo control

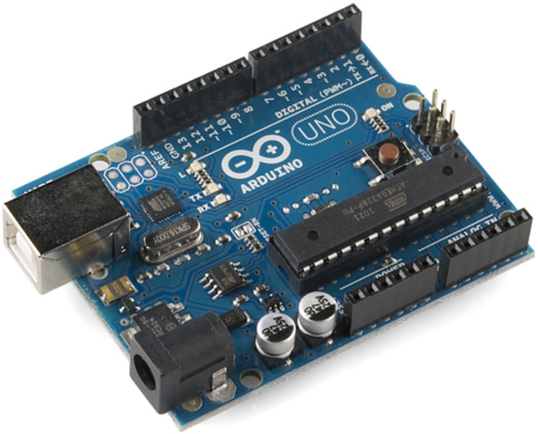
June 24th

Session 6: WAIT A MINUTE

Actions triggering reactions with interrupts

Why Arduino?

- Easy
- Cheap
- Multi-platform
- Large community
- Hard to break
- Lots of 'alternatives'



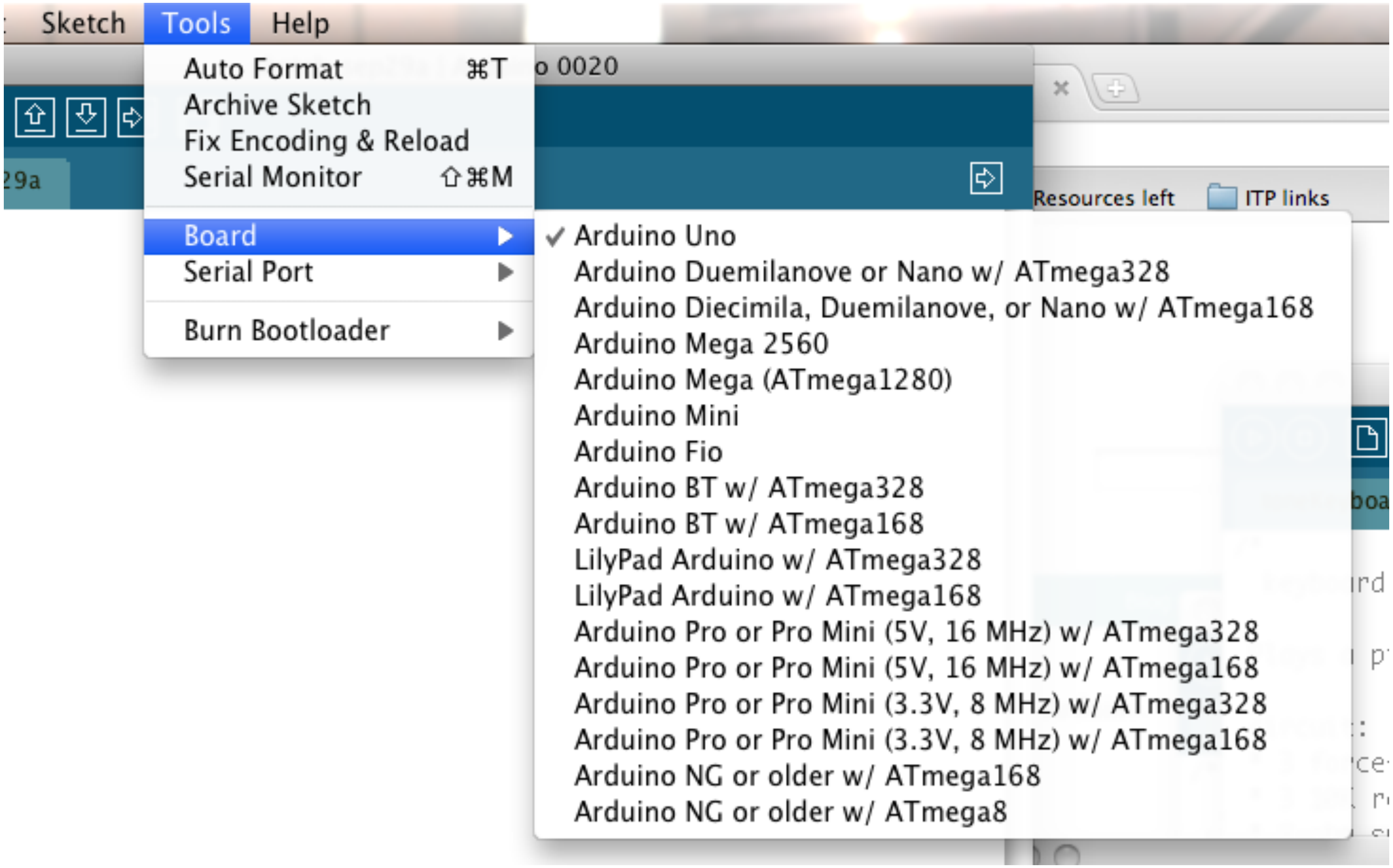
Install Arduino Software

- Arduino Website: http://arduino.cc
- Goto 'Download'
- Download your operating system's version
- Unzip the folder
- Open the folder, click on the 'arduino' executable

Connect Arduino

show windows driver installation process

Select Board



Sketch Tools Help

Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M

Board ▶
Serial Port ▶

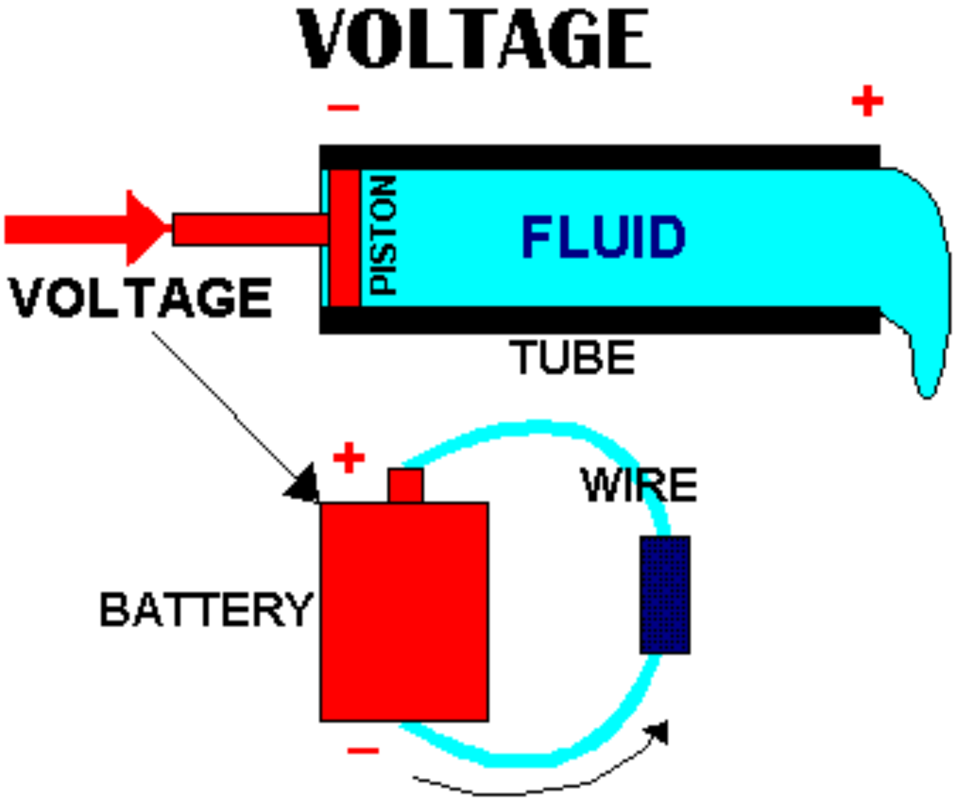
Burn Bootloader ▶

✓ Arduino Uno
Arduino Duemilanove or Nano w/ ATmega328
Arduino Diecimila, Duemilanove, or Nano w/ ATmega168
Arduino Mega 2560
Arduino Mega (ATmega1280)
Arduino Mini
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8

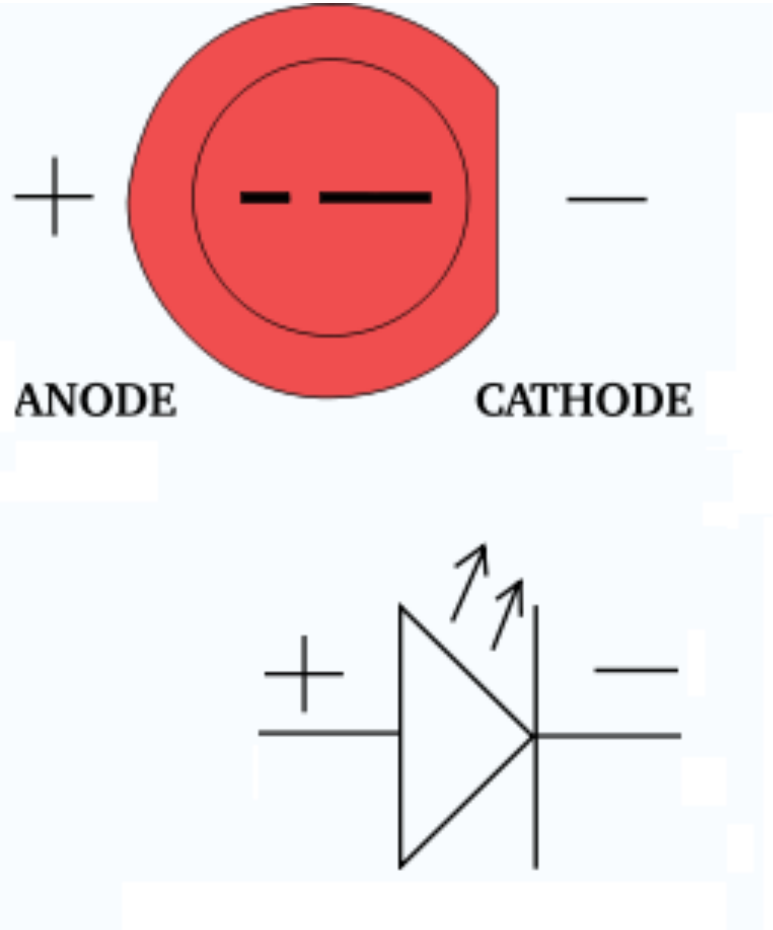
Select Serial Port

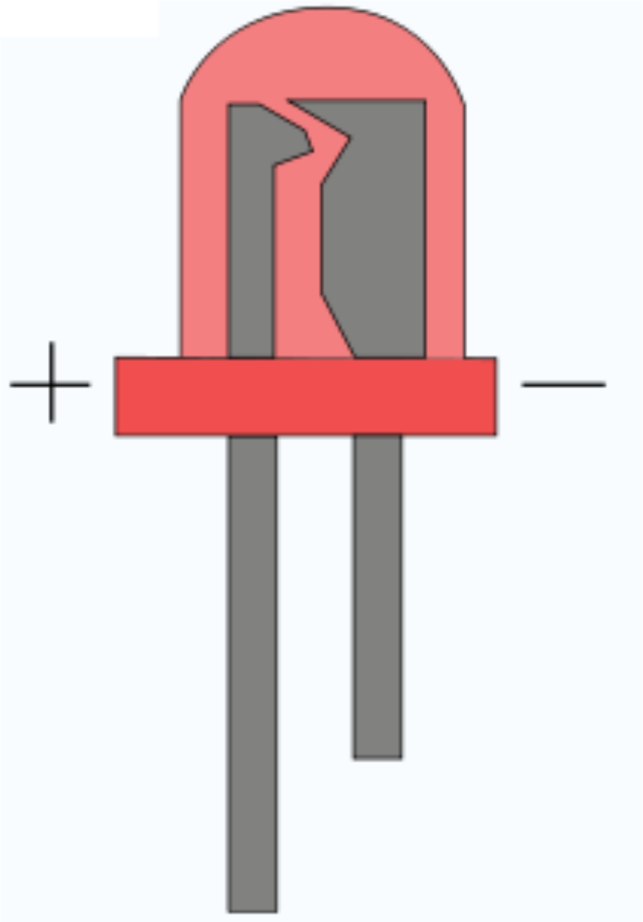
show image

What is Voltage?

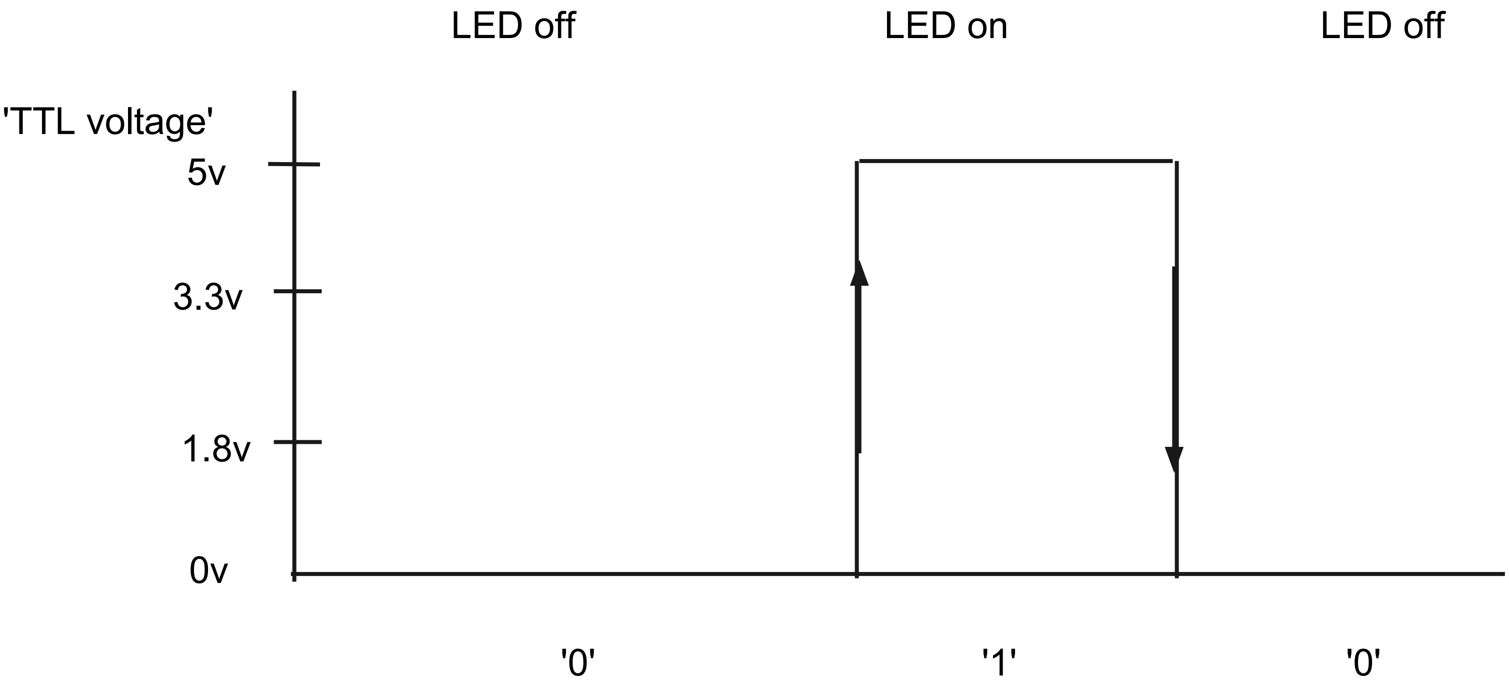


Intro to LEDs

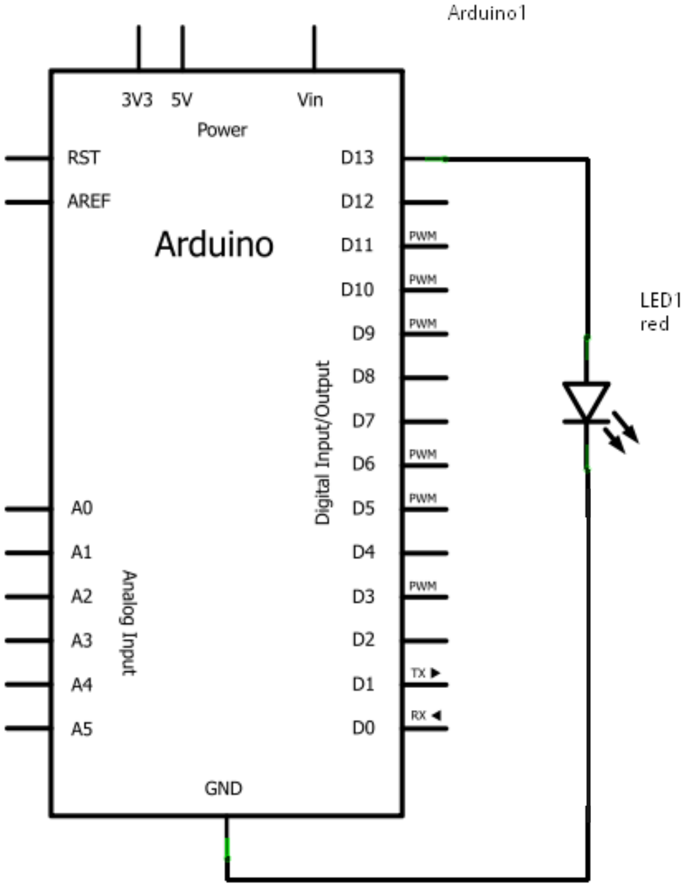


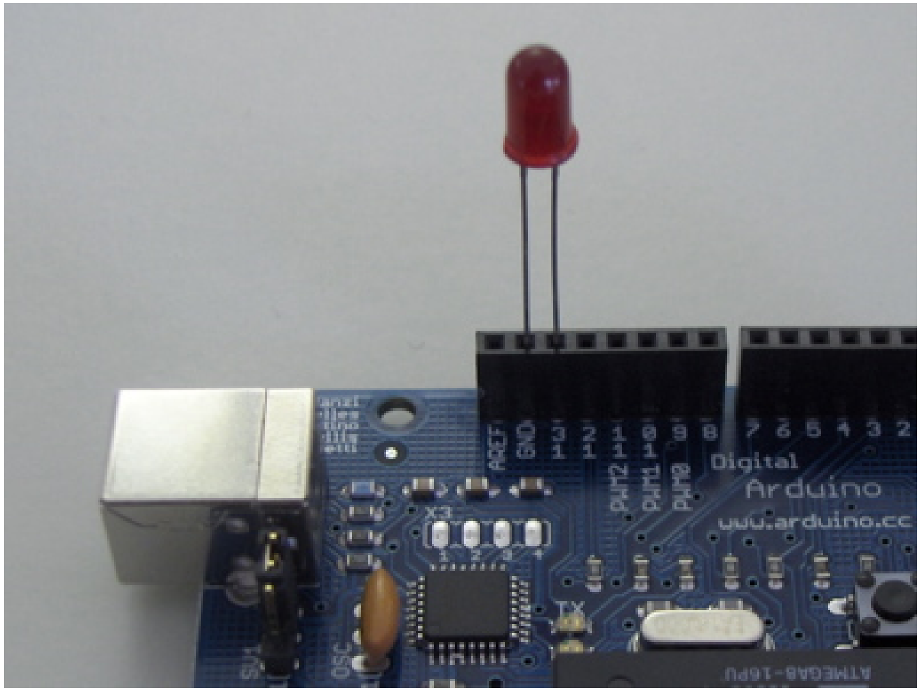


Intro to Digital



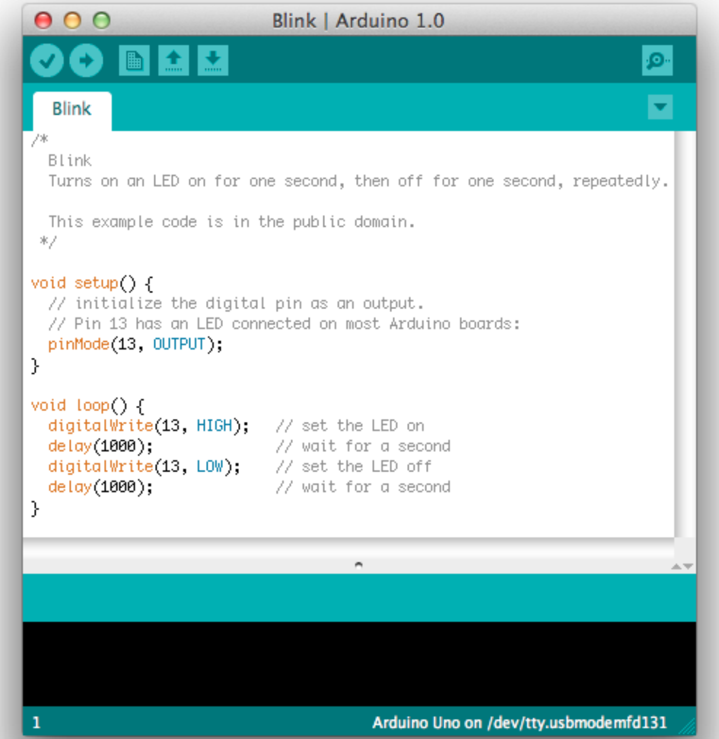
Connect LED


Arduino1
3V3
5V
Vin
Power
RST
AREF
Arduino
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
PWM
TX
RX
Digital Input/Output
A0
A1
A2
A3
A4
A5
Analog Input
GND
LED1
red


AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
PWM2
PWM1
PWM0
Digital
Arduino
www.arduino.cc
X3
TX
ATMEGA8-16PU

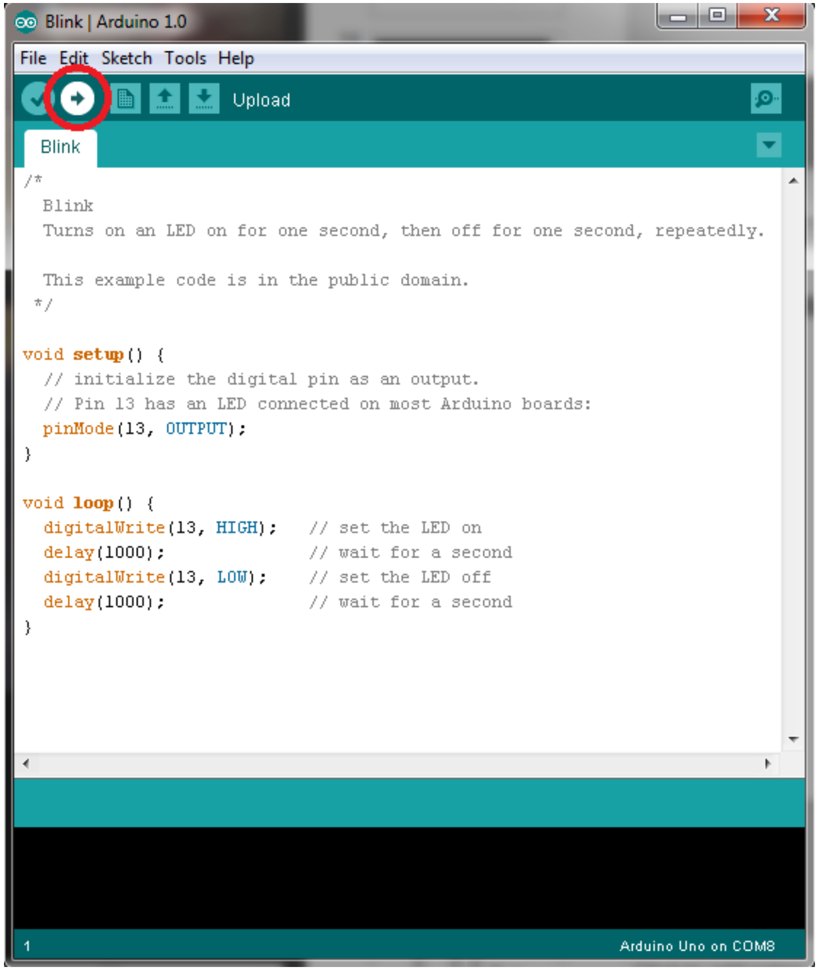
Arduino LED Code

File > **Examples** > **1.Basics** > **Blink**.



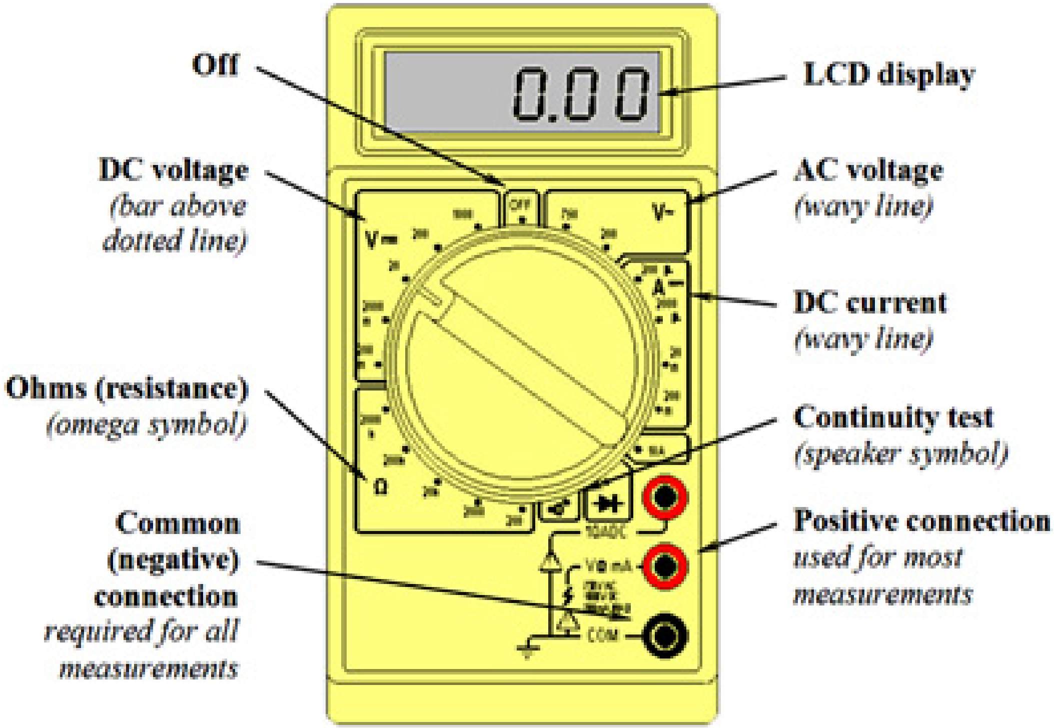
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Upload Code



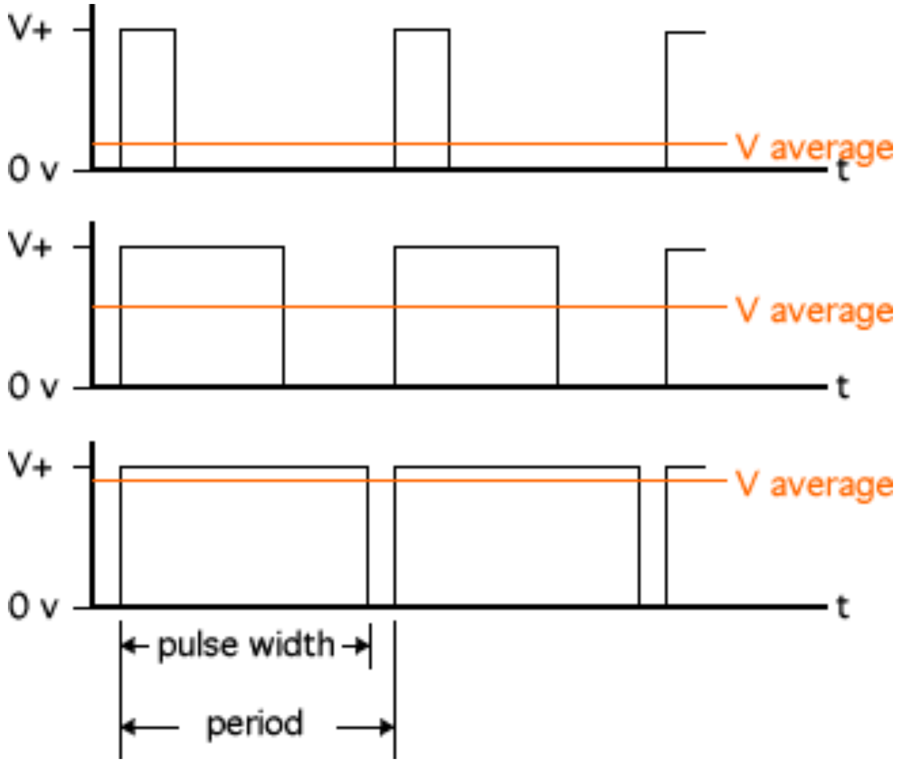
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

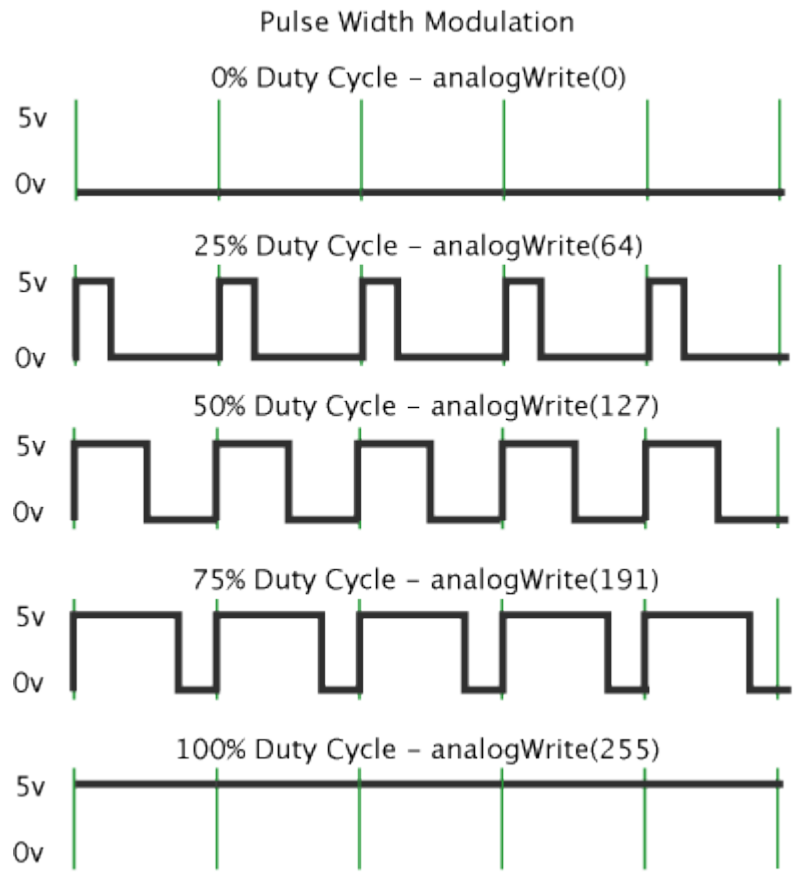
How to use a Multimeter



~Try measuring your LED voltage changing!~

Pulse Width Modulation (PWM)



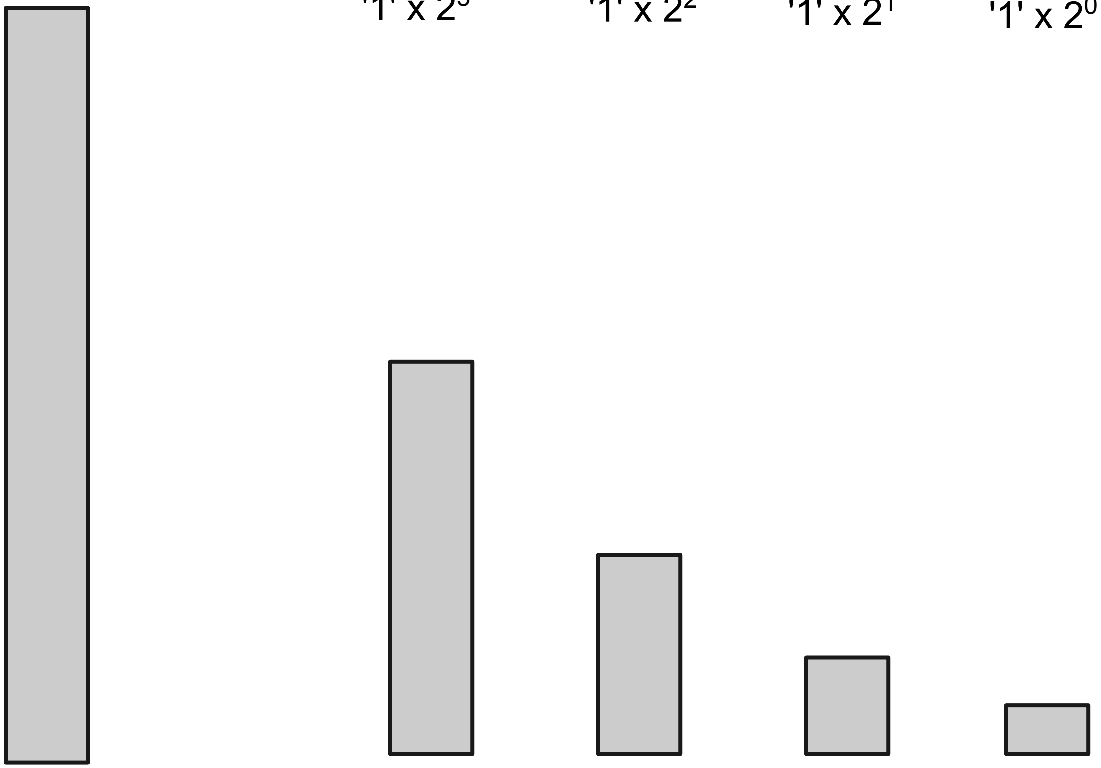

Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v

Intro to Binary

'1111'

15 = 8 + 4 + 2 + 1

'1' x 2^3 '1' x 2^2 '1' x 2^1 '1' x 2^0

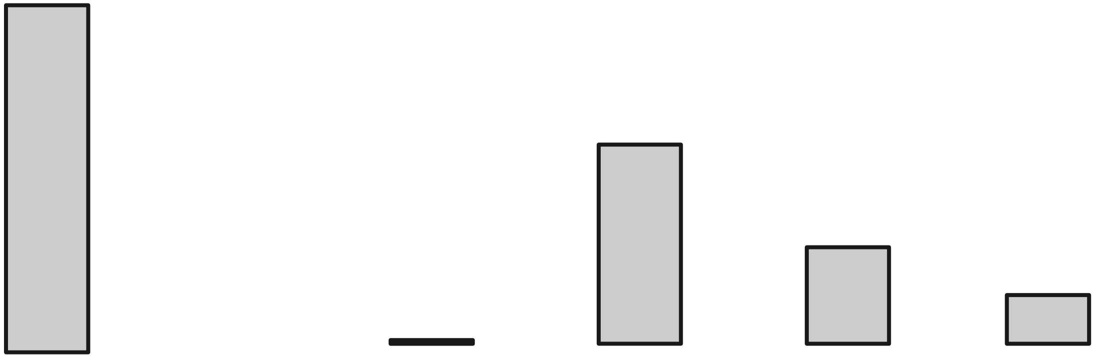


Intro to Binary (continued)

'0111'

7 = 0 + 4 + 2 + 1

'0' x 2^3 '1' x 2^2 '1' x 2^1 '1' x 2^0

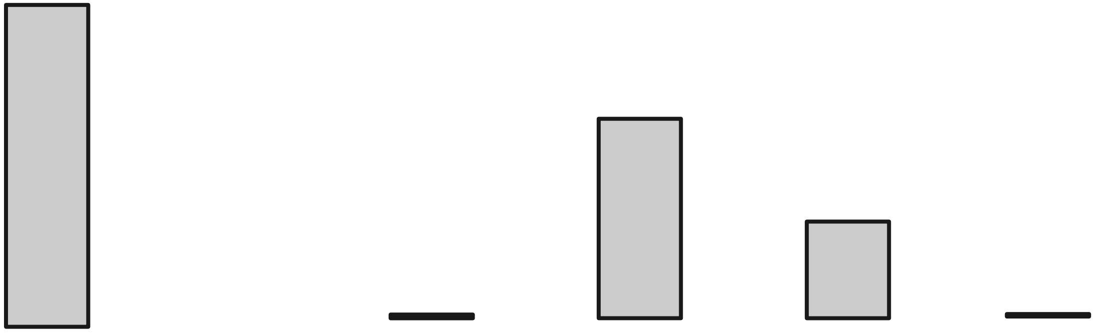


Intro to Binary (continued)

'0110'

6 = 0 + 4 + 2 + 0

'0' x 2^3 '1' x 2^2 '1' x 2^1 '0' x 2^0

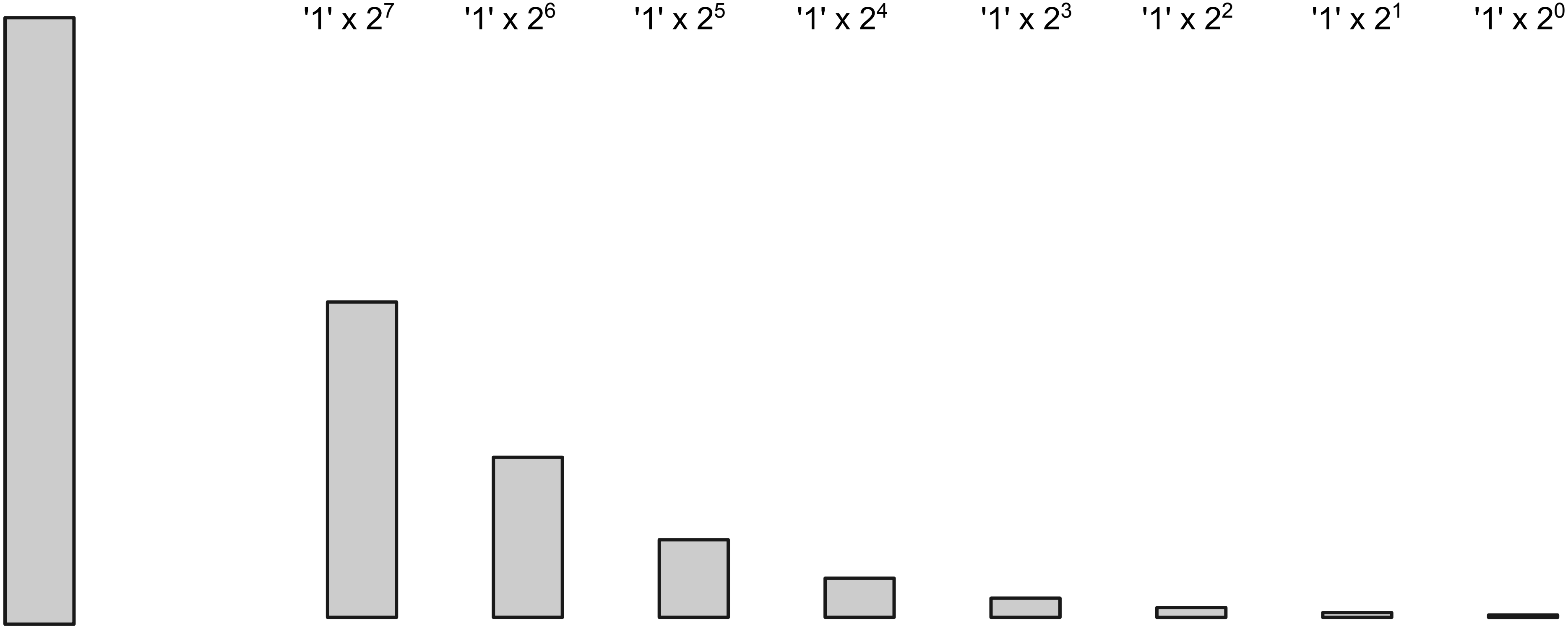


Intro to Binary

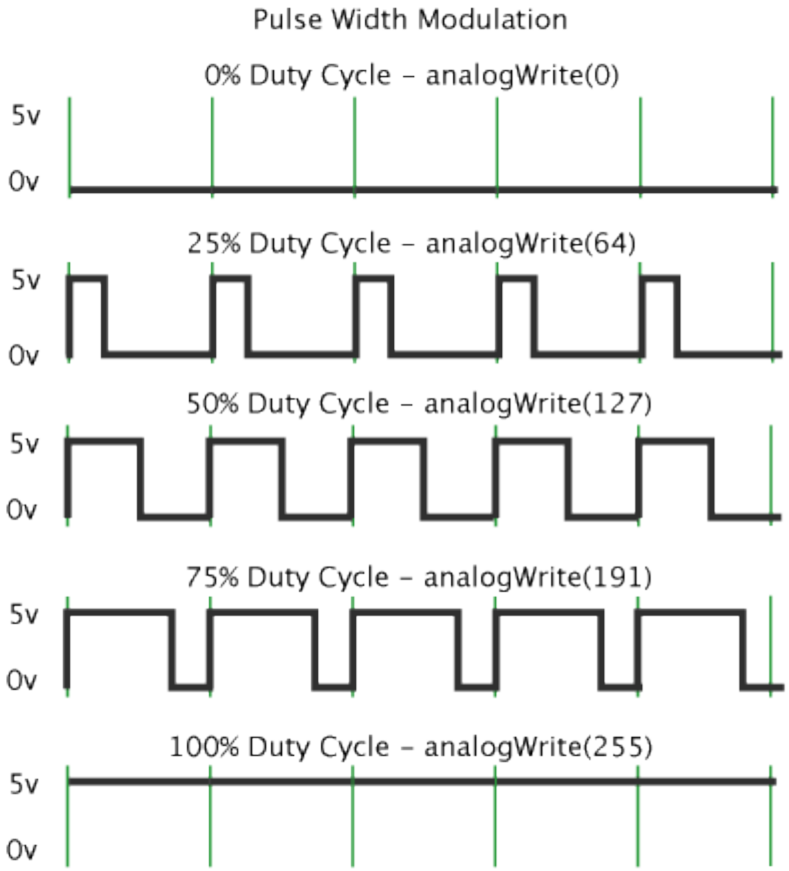
'1111 1111'

255 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1

'1' x 2^7 '1' x 2^6 '1' x 2^5 '1' x 2^4 '1' x 2^3 '1' x 2^2 '1' x 2^1 '1' x 2^0



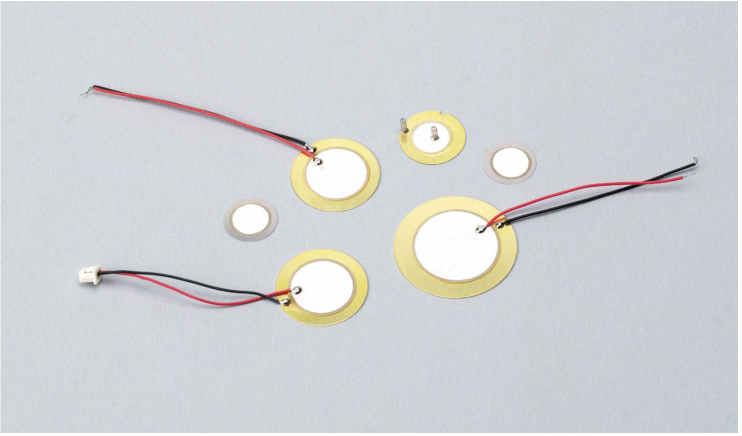
PWM the LED

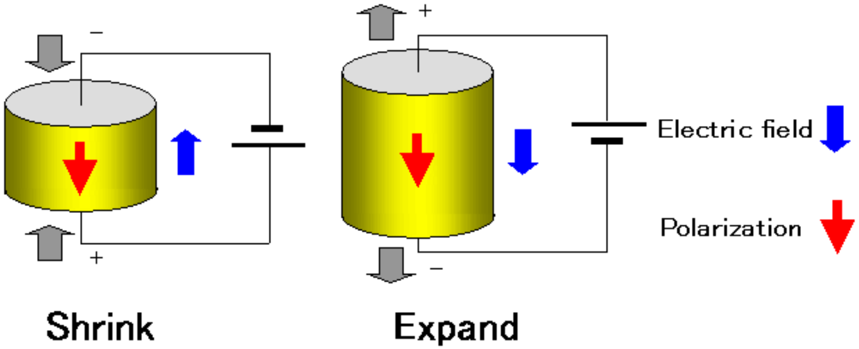


Pulse Width Modulation
0% Duty Cycle - analogWrite(0)
5v
0v
25% Duty Cycle - analogWrite(64)
5v
0v
50% Duty Cycle - analogWrite(127)
5v
0v
75% Duty Cycle - analogWrite(191)
5v
0v
100% Duty Cycle - analogWrite(255)
5v
0v

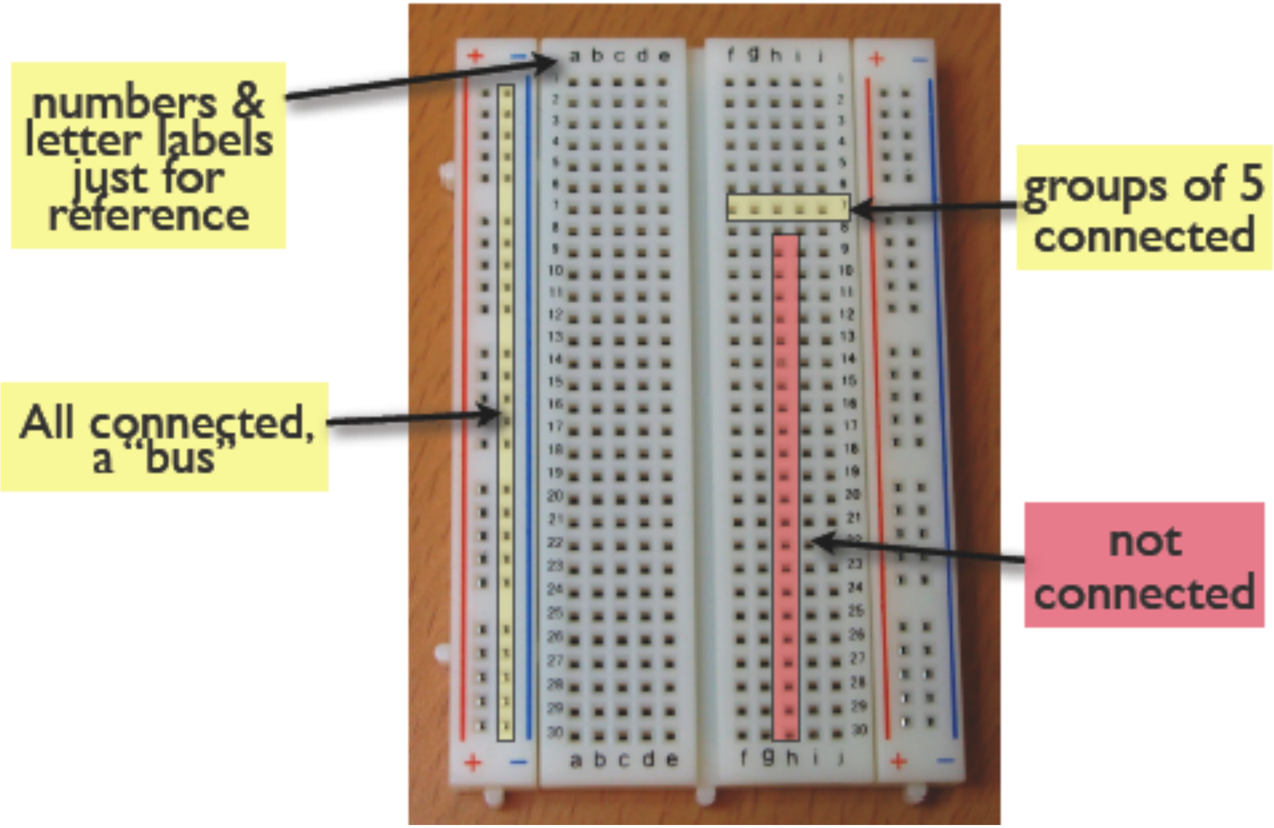
Arduino Code

Intro to Buzzers





Intro to Breadboards



numbers & letter labels just for reference

All connected, a "bus"

groups of 5 connected

not connected

Wiring the Buzzer

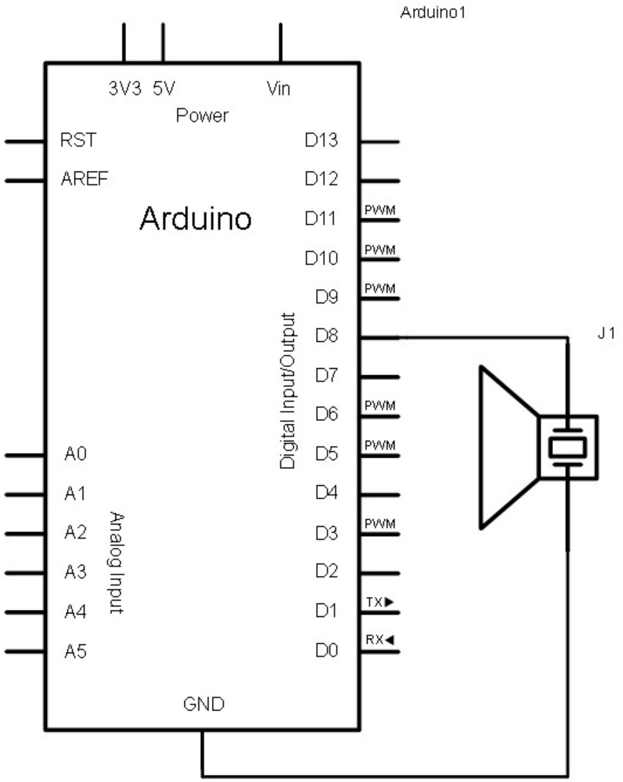


Fig 1 Connect the piezoelectric buzzer to Pin 8 The program will control the buzzer through Pin 8 on the Arduino

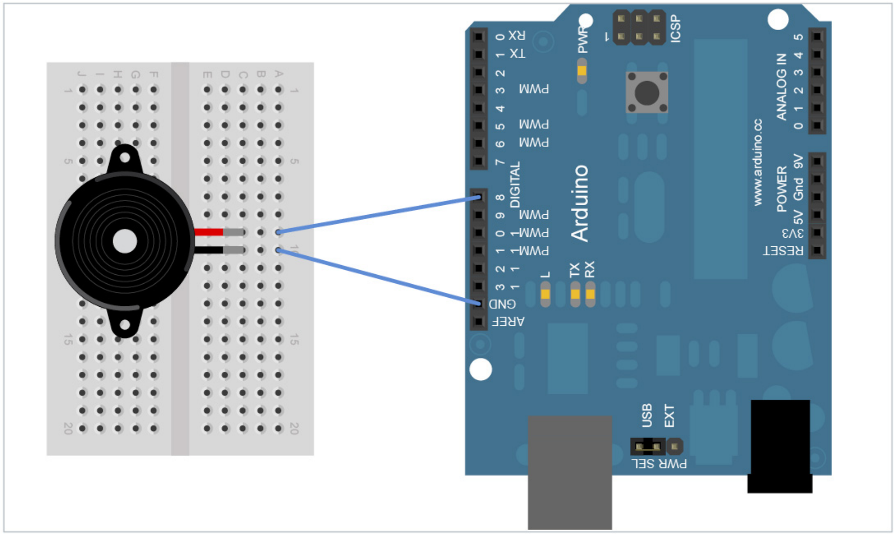
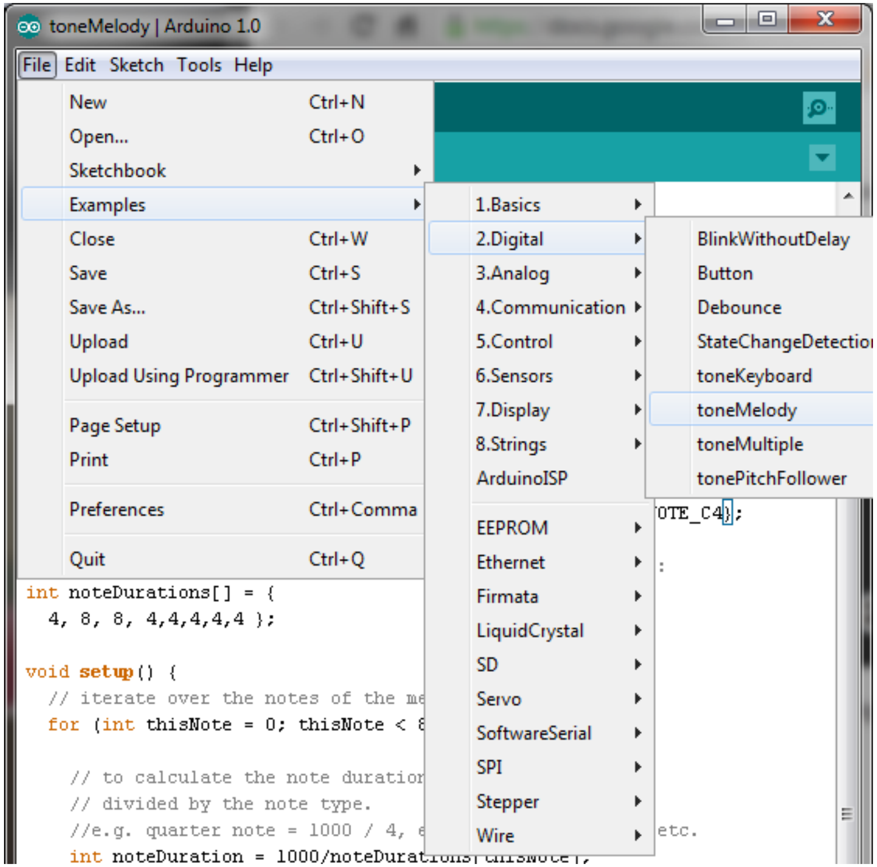


Fig 2 Connect the piezoelectric buzzer to the Ground Pin Completing the circuit

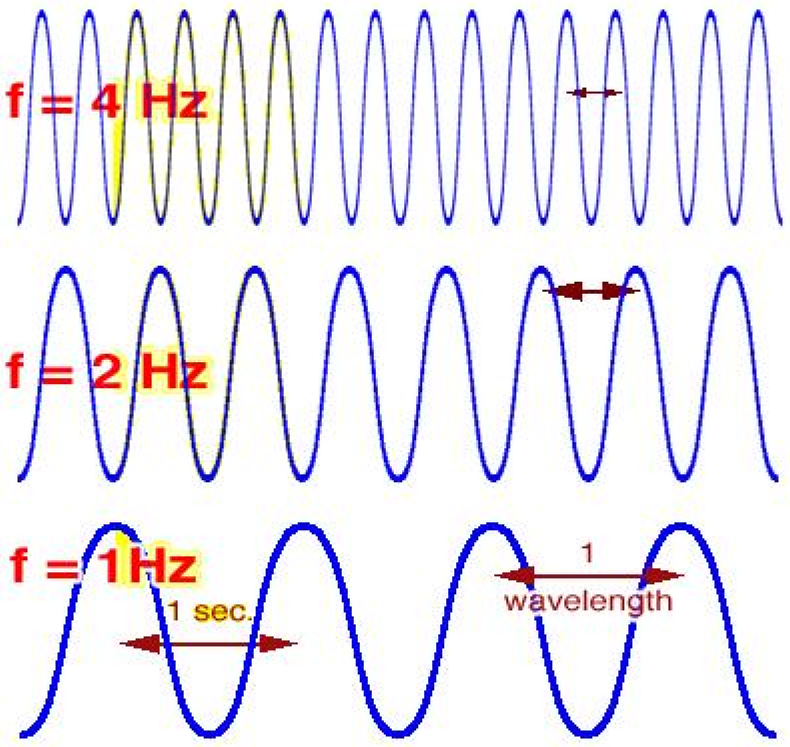
Buzzer Code

File > Examples > Digital > toneMelody



```
OTE_C4};

int noteDurations[] = {
  4, 8, 8, 4,4,4,4,4 };

void setup() {
  // iterate over the notes of the me
  for (int thisNote = 0; thisNote < 8

    // to calculate the note duration
    // divided by the note type.
    //e.g. quarter note = 1000 / 4, e                    etc.
    int noteDuration = 1000/noteDurations[thisNote];
```

Intro to Frequency



http://bbamusic.wikispaces.com/Three+Components+of+Sound

Changing Frequency

updated code