Lab 120

Introduction to Arduino and Electronics

Class 4

14 July 2009 - AS220 Labs - John Duksta

What's for Today

- Recap
- Building an Audio Amp
- Visualizing Sound
- Project Show and Tell!

Recap: Blinky LED

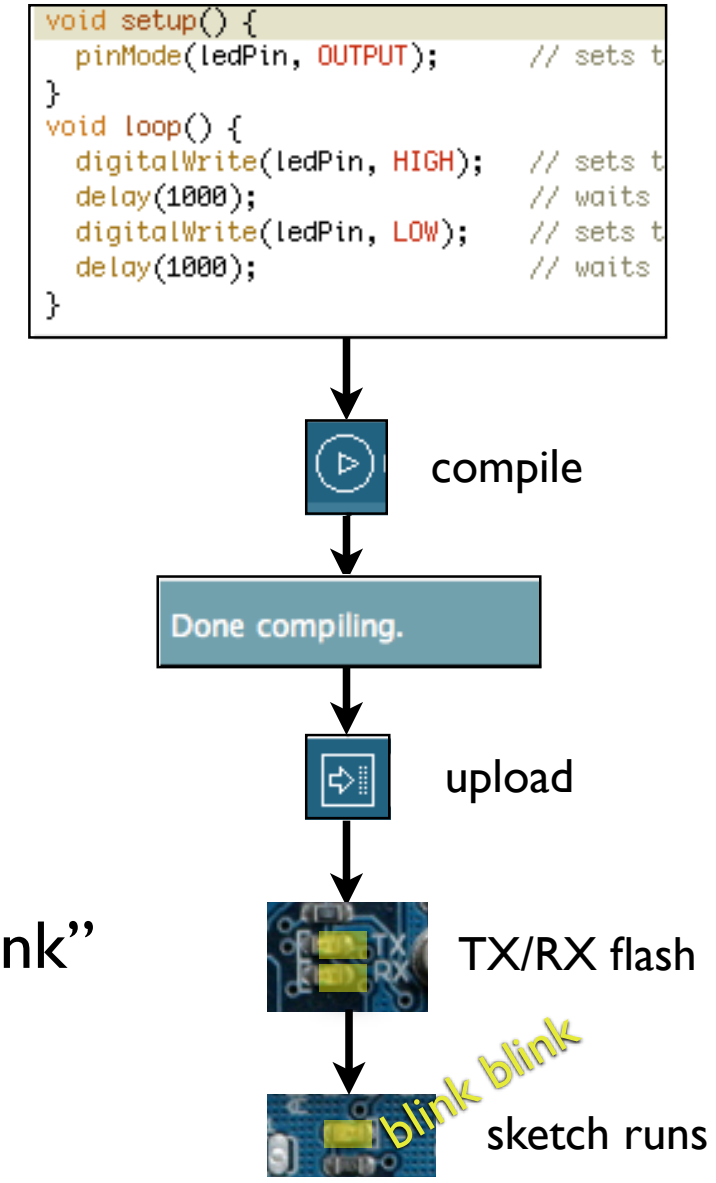
Make sure things still work

```
int ledPin = 13;                  // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

Load “File/Sketchbook/Examples/Digital/Blink”

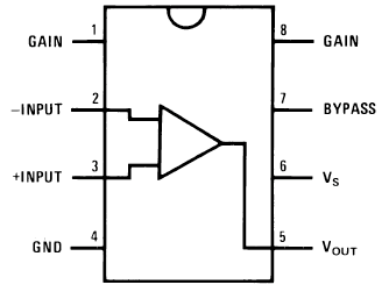


Visualizing Sound

- Use a microphone to capture sound
- Digitize the analog signal
- Send the digitized signal to the computer for visualization with Processing
- Caveat: The signal from the microphone is too weak, use an amplifier to increase the signal for good resolution on the A/D converter

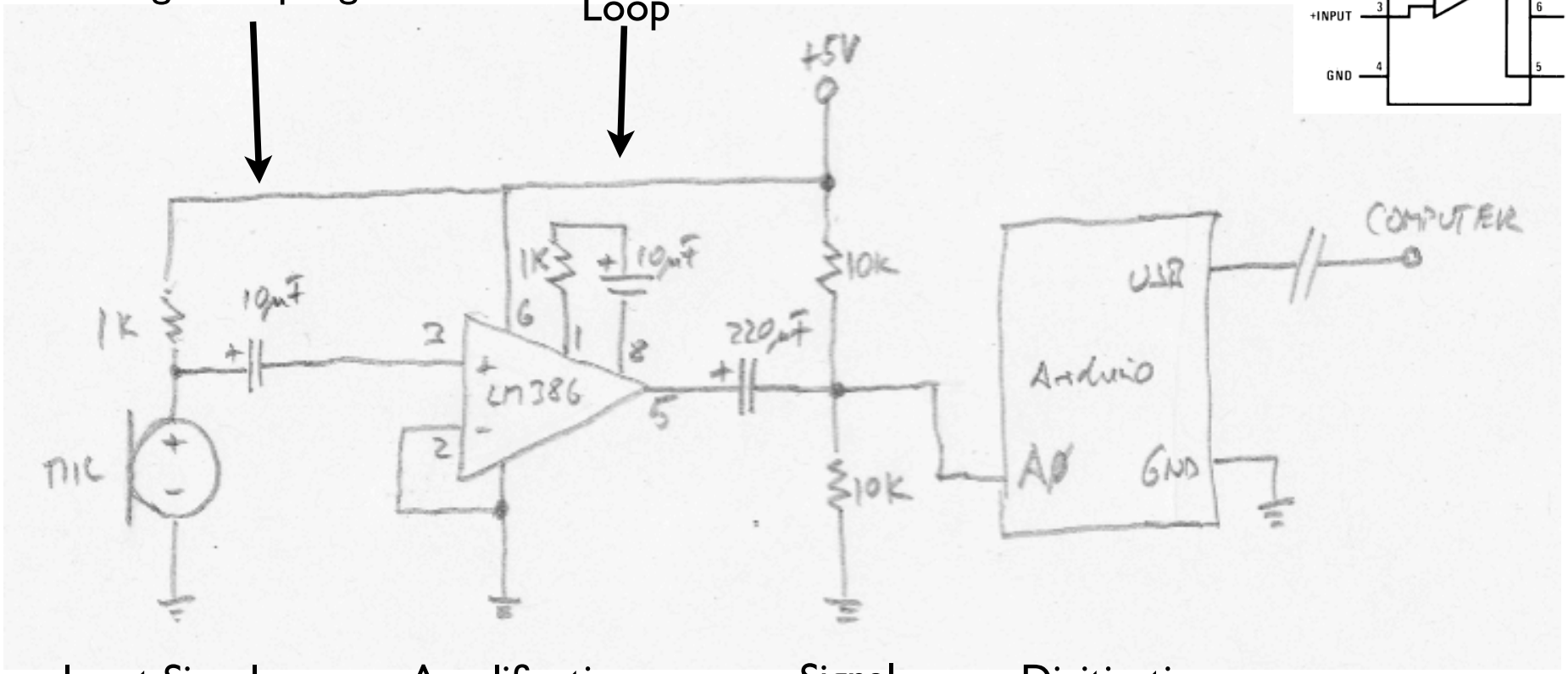
Visualizing Sound

LM386



Stage Coupling

Amp Feedback Loop



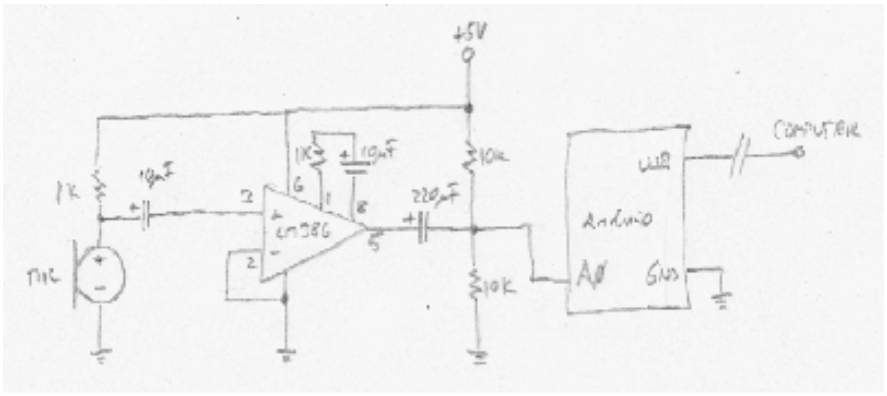
Input Signal

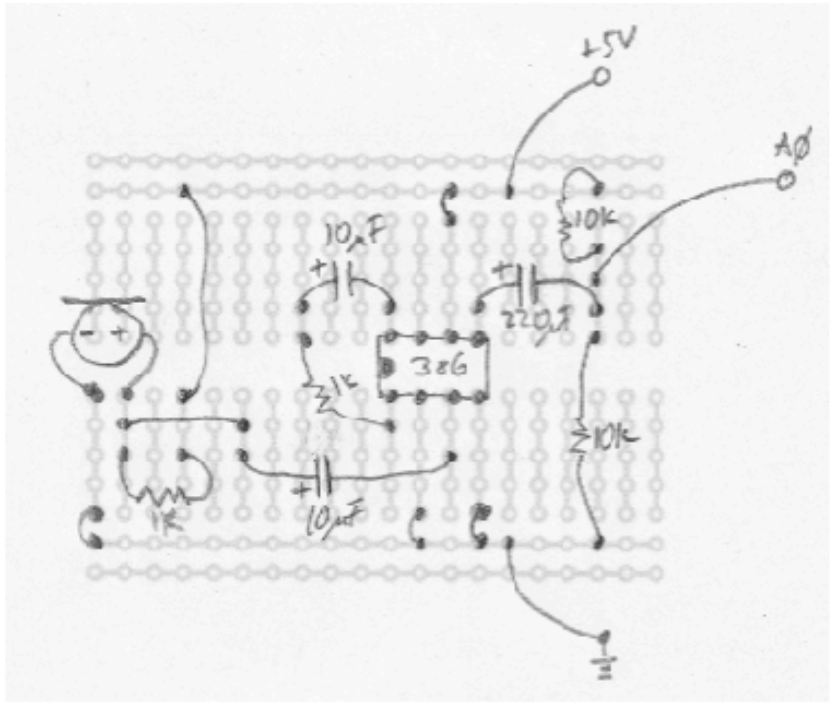
Amplification

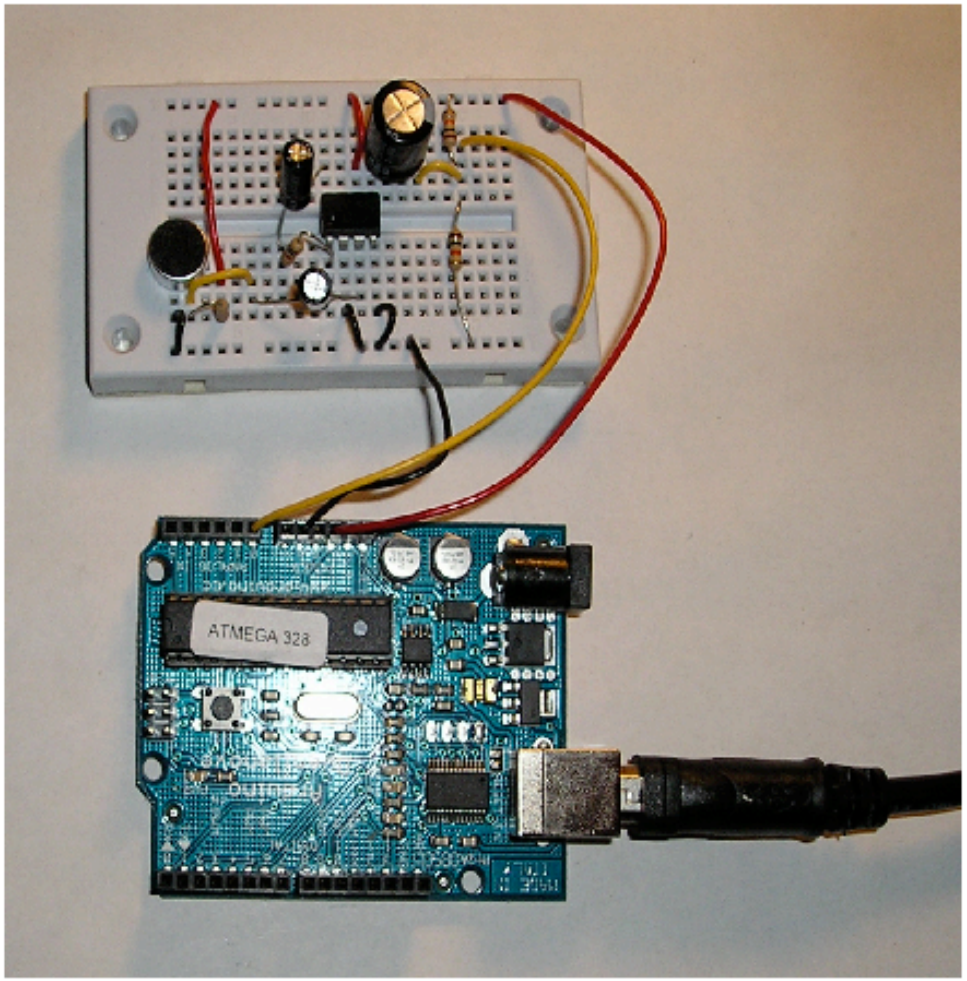
Signal Balancing

Digitization

Visualizing Sound







Visualizing Sound

```
// visualizing sound – Arduino Sketch
// read input from the amplified mic
// and send the value to the multi-media
// computer for visualization

int soundinPin = 0;
int val = 0;

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

void loop() {
  val = analogRead(soundinPin);
  val = map(val,0,1023,0,255);
  Serial.print(val,BYTE);
  delay(8);
}
```

```
// Visualize sound - Processing sketch
// visualize input from the arduino board
// based on Graph by David A. Mellis

import processing.serial.*;

Serial myPort;
int inputByte = 0;
// Store the last 64 values received so we can graph them.
int[] values = new int[64];

void setup()
{
  size(512, 256);
  frameRate(120);
  myPort = new Serial(this, "COM6", 9600);
}

void draw()
{
  background(53);
  stroke(255);

  // Graph the stored values by drawing lines between them.
  for (int i = 0; i < 63; i++){
    line(i*8,255-values[i],(i+1)*8,255-values[i+1]);
    // Shift over the existing values to make room
    values[i] = values[i+1];
  }
  if (myPort.available() > 0) {
    inputByte = myPort.read();
    values[63] = inputByte;
  }
}
```

Visualizing Sound

- A big problem with this application is that the display has very low bandwidth
- On most computers you can probably not achieve a frame rate higher than 240 frames per second
- This means we sample our sound wave every 4 msec (or sample freq=240Hz)
- This implies that the maximum frequency that we can visualize without distortion is 120Hz, not very useful

Visualizing Sound

- Instead of visualizing the sound wave, visualize the composition of sound in terms of frequencies
- Fast-Fourier Transform (FFT)
- In this case we turn our Arduino board into a DSP chip
- However, the code for this is too complex to present here.

The Clapper

- Leave the hardware as is but we change the software
- If we hear a loud noise send a signal to the computer
- On the computer the signal determines how fast a line rises on a display

The Clapper

```
// The Clapper – Arduino Sketch
// read input from the amplified mic,
// if you hear a loud sound
// send a signal to the computer

// read the sound input on analog pin 0
int soundinPin = 0;
// the sound threshold above which we
// consider a sound to be loud
int threshold = 700;
// the value read from the soundinput
int val = 0;

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

void loop() {
  val = analogRead(soundinPin);
  if (val >= threshold) {
    Serial.print(1,BYTE);
    // debounce
    delay(10);
  }
}
```

Note: Good example of signal thresholding.

Note: Clapping produces echoes etc, we debounce our signal by waiting until the echoes are gone so we don't accidentally react to the echoes.

```
// The Clapper – Processing Sketch
// Every time we receive an event on the serial
// port we let the line rise faster up to a certain
// value and then we start again.

import processing.serial.*;
Serial myPort;
int inputByte = 0;
float y = 100;
int increment = 1;

void setup()
{
  size(200, 400);  // Size should be the first statement
  stroke(255);     // Set stroke color to white
  frameRate(60);
  // need to pick the right com port
  myPort = new Serial(this, "COM6", 9600);
}

void draw()
{
  background(0);   // Set the background to black
  line(0, y, width, y);
  y = y - increment;
  if (y < 0) {
    y = height;
  }
}

void serialEvent(Serial p) {
  // remove the byte from the serial port
  inputByte = p.read();
  // speed up the line
  increment = (increment + 2) %  8;
  println(increment);
}
```

More Processing

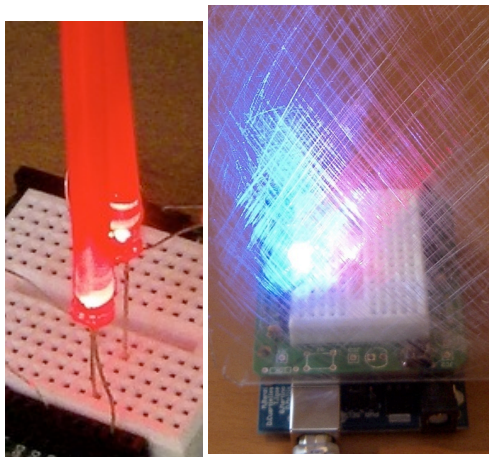
- Processing has many other libraries
- Quicktime
- Sound
- OpenGL
- Network (Check out "Talking to the Cloud")

Take a Break

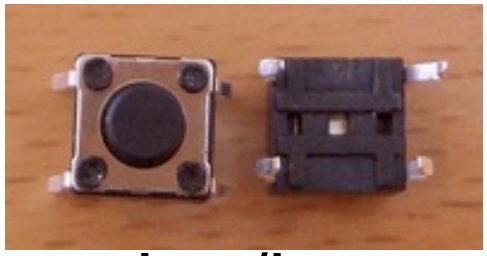
Project Show and Tell!

Summary

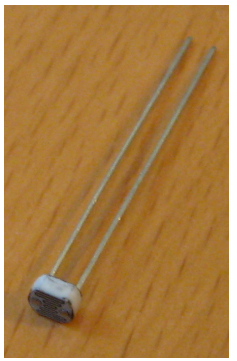
You've learned many different physical building blocks



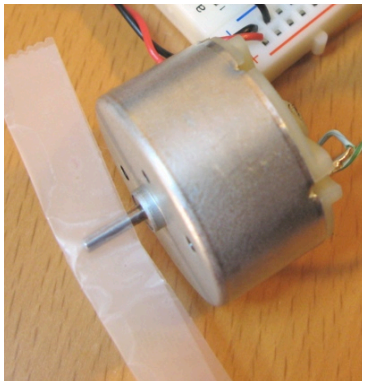
LEDs



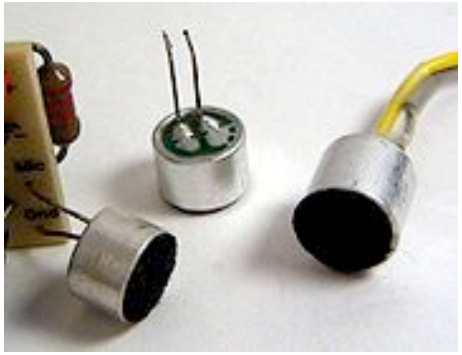
switches/buttons



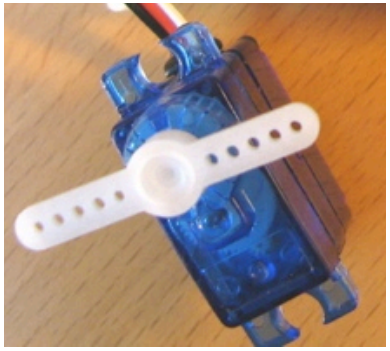
resistive sensors



motors



microphones



servos

Summary

And you've learned many software building blocks

pulse width modulation

serial communication

analog I/O

digital I/O

data driven code

frequency modulation

multiple tasks

Summary

Hope you had fun and continue playing with Arduino

Feel free to contact me to chat about this stuff

END Class 4

http://duksta.org/arduinoclass

John Duksta

john@duksta.org

Giving Credit

This courseware is a mashup of Tod E. Kurt's Bionic Arduino course, taught at Machine Project in LA and Lutz Hamel's Intro to Arduino course taught here at AS220