

3.5 hour Drawing Machines Workshop

SIGGRAPH 2013 – Educator’s Focus

Sponsored by the SIGGRAPH Education Committee

Overview: The workshop is composed of three hands-on activities, each one introduced with a short presentation by one of the instructors. The time schedule is approximate, and may drift, but shows how quickly we’ll have to move to get through everything!

1:30-1:40: Introduction of instructors, and of students – forming teams

Instructors:

Erik Brunvand, School of Computing, University of Utah (elb@cs.utah.edu)

Ginger Alford, Trinity Valley School, Ft. Worth, TX (gralford@acm.org)

Paul Stout, Dept. of Art and Art History, Univ. of Utah (paul.stout@gmail.com)

1:40-1:50: Introduction to drawing machines (Erik)

1:50-2:30: Activity – basic programming, wiring, and LEDs

- Powerpoint/handouts
 - o Intro to programming / Arduino
 - o Intro to electronics
 - o How LEDs work
 - o Connections to Arduino (proto-boards, digital outputs, etc.)
- Activity:
 - o Wire up external LED to Arduino
 - o Upload and run Blink (Arduino IDE sanity check)
 - o Fading LED with analogWrite

2:30-2:40: Short break

2:40-2:50: Introduction to kinetic art (Paul)

2:50-3:30: Activity – basic sensors and servos

- Powerpoint/handouts
 - o Intro to servos (wiring and programming)
 - o Intro to sensors (knobs and light sensors)
 - o Intro to calibration (serial output)
- Activity:
 - o Wire up a sensor (potentiometer)
 - o Wire up a servo

- Modify existing program to make servo move in response to the sensor (turn knob – turn servo)

3:30-3:40: Short Break

3:40-3:50: Applications to high school classes (Ginger)

3:50-4:30: Activity – student-designed drawing machines

- Use servos, potentiometers (and LEDs if you like)
- Use foam-core, tape, pens
- Make something that draws!
- Copy example machines, or design your own...

4:30-5:00 Demos!

Contents of Kit (one per team) (please return after the workshop)

- Arduino microcontroller (either Duemilanove or Uno)
- Solderless breadboard
- Wires for connecting components
- Two standard 180° servos
- Two potentiometers (blue plastic)
- Assorted LEDs
- 220Ω resistors (red red brown stripes)
- Two CDS light sensors (squiggly lines on flat top)
- 10kΩ resistors (brown black orange stripes)
- USB cable
- Masking tape

Shared supplies

- Foam core board
- Box cutters for cutting foam core (please cut on masonite cuttings boards)
- Pens, pencils, markers, etc.
- Extra servos
- Extra wires
- Extra LEDs
- Extra resistors
- Extra potentiometers
- Extra light sensors
- Assorted screws, washers, nuts, etc.
- Drawing papers rolls (cut off 2ft sections for making drawings)
- Wood blocks for drawing machine supports
- Extension wires for connecting servos to breadboard

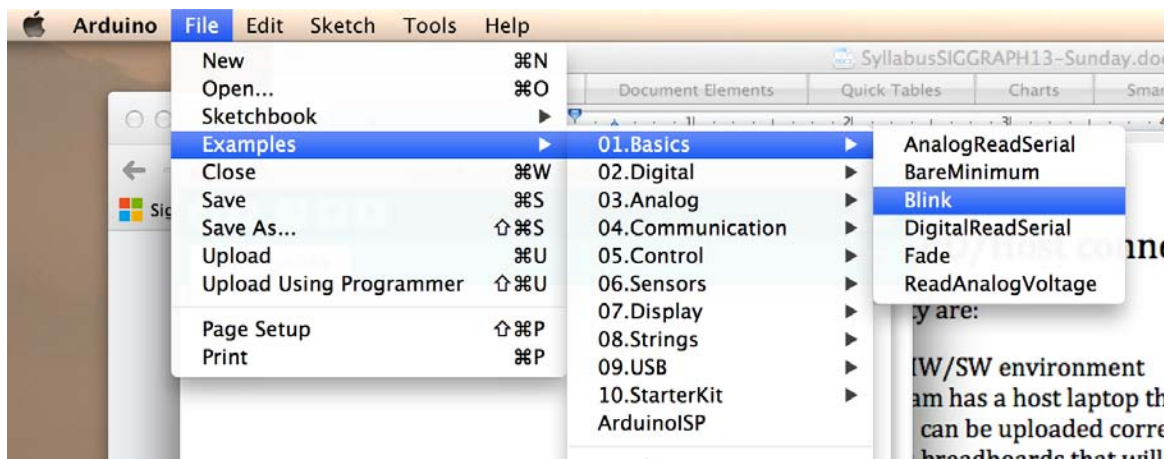
Activity #1: Arduino/LED/Host connection

The main points of the first activity are:

- to introduce the Arduino HW/SW environment
- to make sure that every team has a host laptop that has the IDE installed correctly so that programs can be uploaded correctly to the Arduino board
- to introduce the solderless breadboards that will be used to connect external electrical components
- to introduce the LED and the electrical specifications that go with that type of component

The activity will proceed as follows (instructors will guide you through these steps):

1. Form teams of two or three people sharing a Drawing Machines kit
2. Make sure that the Arduino software is loaded onto a laptop for each team
3. We will introduce the contents of the kit to the whole group. You will follow along as a team and identify each of the components
4. Each team will load the “Blink” program from the Arduino IDE, upload it to their boards, and verify that the tools are working for every team.



```
Blink | Arduino 1.0.1
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

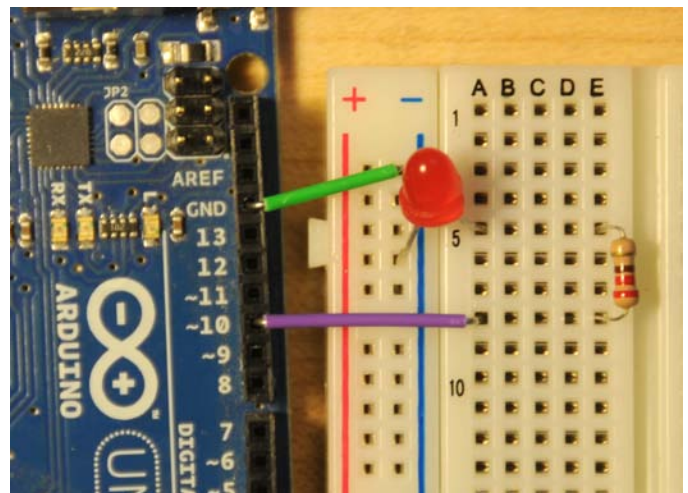
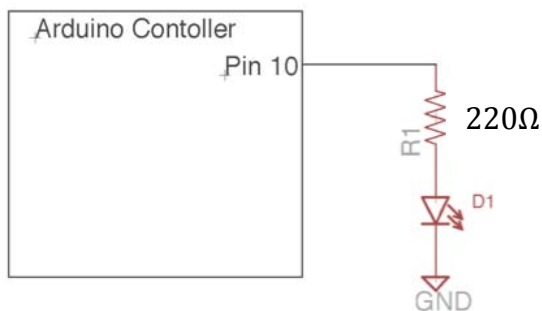
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

1 Arduino Duemilanove w/ ATmega328 on /dev/tty.usbserial-A9007TW7
```

5. Each team will use the solderless breadboard to wire an external LED to their Arduino board
 - a. Re-run Blink with the external LED to verify the connection
 - b. Teams should modify the “delay” statements in the Blink program and note the effect. See powerpoint slides for details.



6. Load the Fade program to use PWM to fade the external LED using `analogWrite()`; (make sure to modify the program to use the pins you’re connected to)

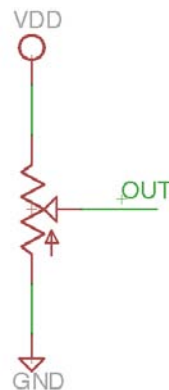
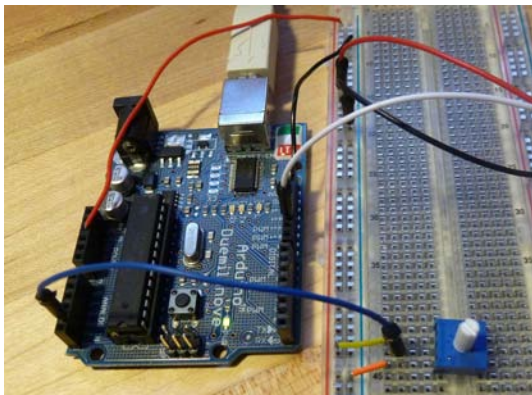
Activity #2: Sensing Analog Voltages, and Servos

The main points of the second activity are:

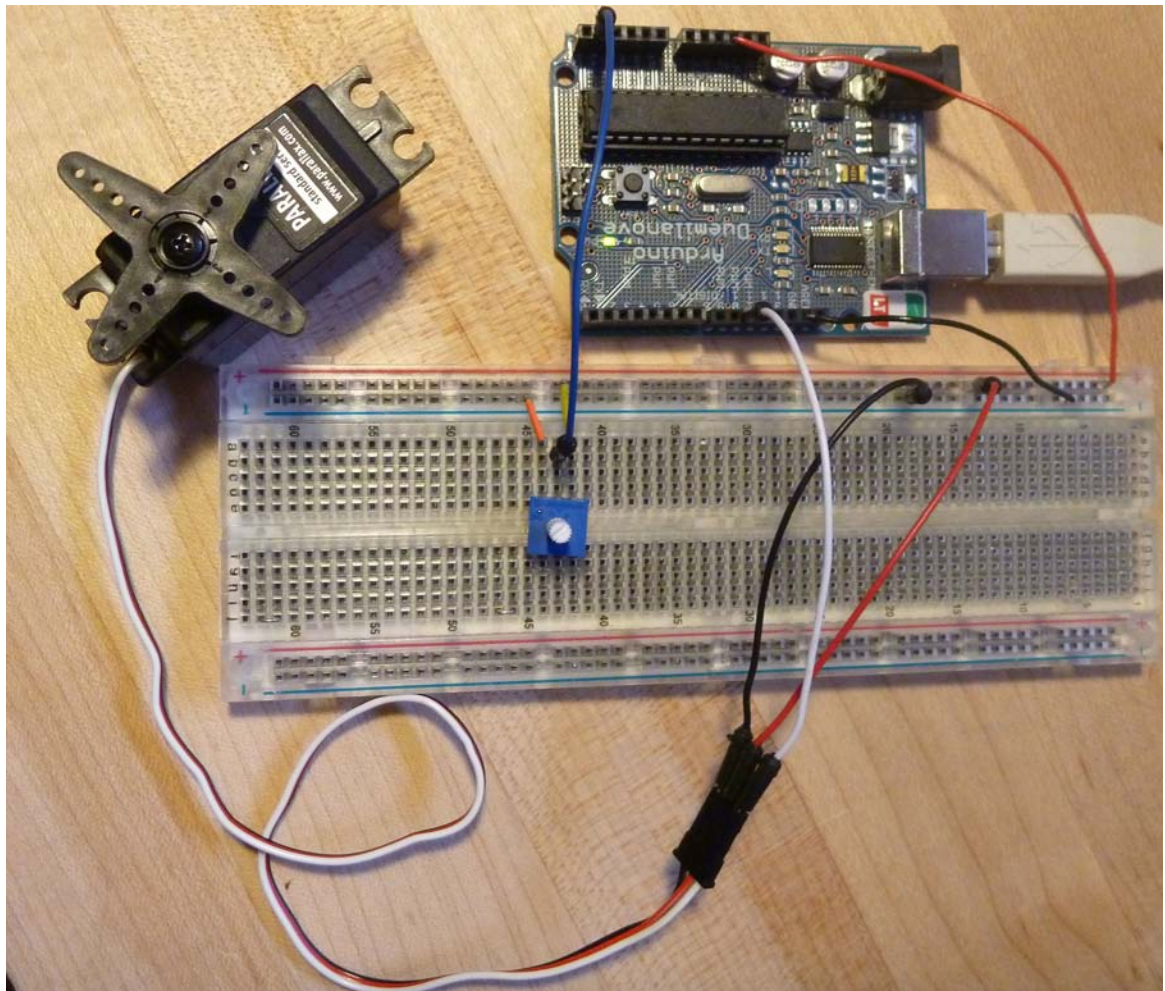
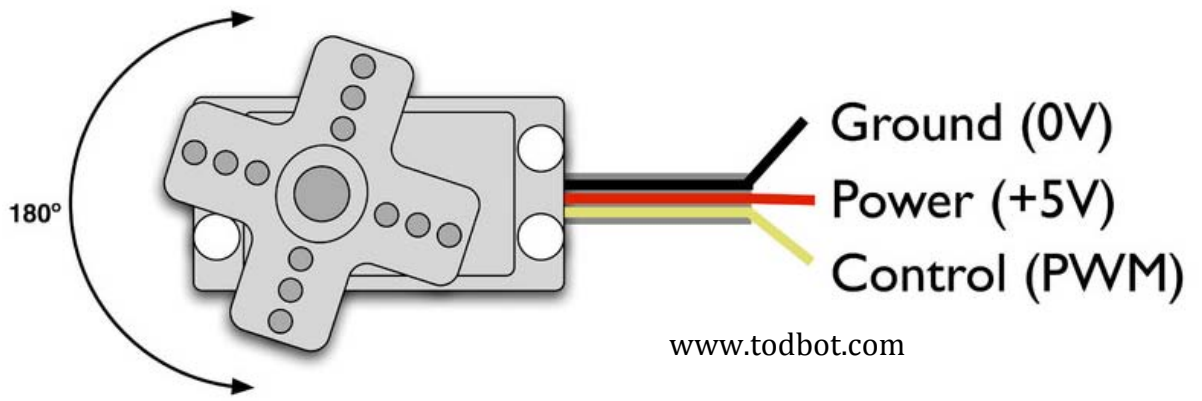
- *to understand analog sensing through an ADC*
- *to understand potentiometers (knobs) and other resistive sensors, and how they're connected to Arduino*
- *to learn about servos and how they're driven by Arduino*
- *to wire up a pot and servo so that the position of the servo is controlled by the pot*

The activity proceeds as follows:

1. Examine the pots in your kit. Note that they have three connection points: the outside connections go to power and ground, and the middle connection point goes to one of the analog input pins on the Arduino
2. Wire up a pot as an external component on your breadboard and connected to the Arduino.



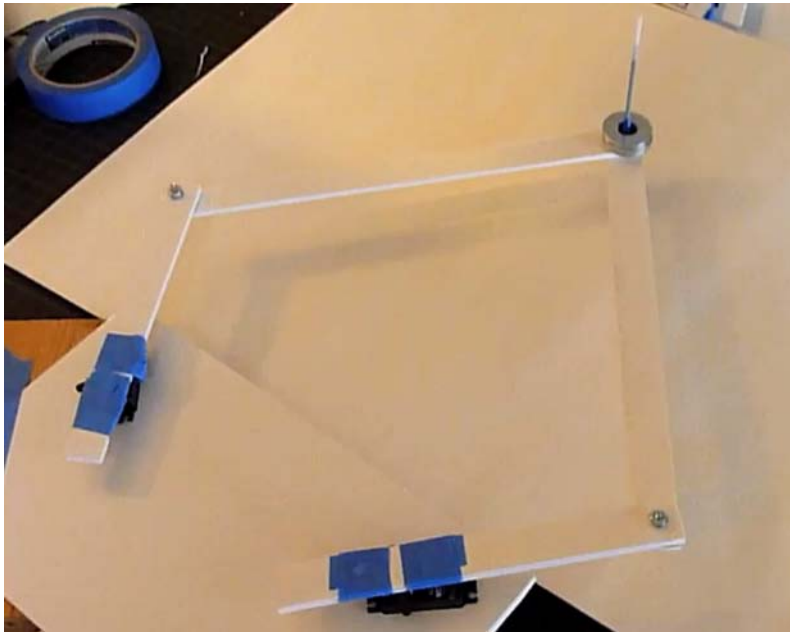
3. Load the potFade program from the DM examples and use the knob to make the external LED from Activity #1 fade as you turn the knob.
4. Connect a servo to your Arduino (using the breadboard for power and ground).
5. Use the potFade program to drive the servo. The knob position now makes the servo move.
6. If time permits, investigate the CDS light sensor as another type of resistive sensor...



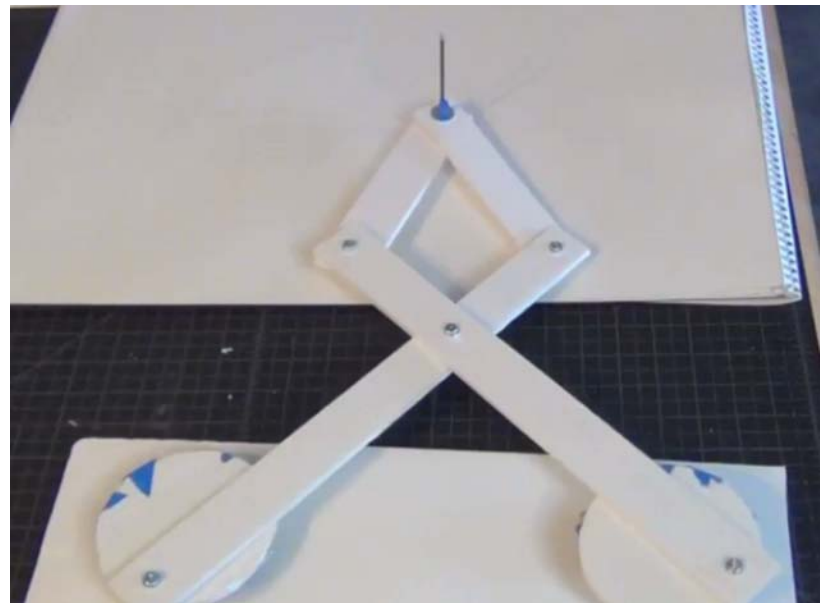
Activity #3: Drawing Machine!

The main point of the third activity is the main point of the workshop: Make a drawing machine!

1. As a team, decide how you want to make your drawing machine. We suggest either the “dancing two-leg machine” or the “harmonograph.” If you choose the harmonograph, you’ll need to substitute continuous rotation servos for the standard servos in your kit



2. Cut foam core pieces for your machine. You’ll likely need to cut holes for the servos, and strips for the drawing machine “arms.”
3. Poke holes in foam core with the screws, and use screws, nuts and washers to make hinge joints.
4. You can use the pen as the “hinge” for the drawing connection.
5. Tape things together, wire up pots and servos, and make a masterpiece!



Description of Example Programs

Programs for Activity One (lighting up LEDs)

Blink: The most basic Arduino example – make an LED blink. Demonstrates basic Arduino code organization and digital output functionality (digitalWrite).

eight-led – these are examples of making a pattern appear on eight LEDs connected to eight digital pins. Each example makes the same pattern, but introduces a different programming concept.

eight-led/example1: Brute force program to make a pattern of lights on eight LEDs.

eight-led/example2: Same pattern on the eight LEDs, but using “for” loops.

eight-led/example3: Uses a separate function to set the values of the LEDs. That function is then called to make the pattern on the LEDs.

eight-led/example4: Introduces arrays to hold the values of the pins that the LEDs are connected to.

eight-led/example5: Introduces binary notation to hold the state of all LEDs rather than individual 1/0 for each LED. This includes unpacking a byte to get each bit value.

Candle: Introduces the “random” function and the analogWrite function to make an LED flicker like a candle.

Candle-small: Shows how the random function can be used directly in the argument list of other functions.

Programs for Activity Two: Inputs, Sensors, and Servos

SimpleButton: Introduces digitalWrite and buttons as input.

ButtonDelay: Use a button to control the blink rate of an LED. Introduces “if” statements based on input data.

PotFade: Uses a resistive sensor (potentiometer/knob) to get a range of values using the analog inputs. Use this to control the brightness of an LED. Also works with a CDS light sensor.

HelloWorld: Shows how to write/print to the “serial monitor” on the host machine. This is how your Arduino can communicate information to the user.

Calibration: Use the serial monitor to see the range of values your sensor is returning. Use that information and the “map” and “constrain” functions to get full range information out of a resistive sensor.

BlinkRate: Use a (calibrated) resistive sensor to control the blink rate of an LED.

SimpleServo: Use a resistive sensor (knob or light sensor) to control the position of the servo arm. Introduces the servo library for moving a servo arm, and also the use of classes and methods.

ServoCalibration: Add calibration to SimpleServo so that you can get full-range motion from a servo.

Activity three: use inputs (buttons, sensors, knobs) and outputs (LEDs and servos) to make a machine that makes drawings!

ServoTemplate: Not a full program, but an example of how to connect and control three separate servos.

Equipment List and Example Suppliers

This is a definitely non-exhaustive list of some of the equipment used in this workshop, a rough guide to prices, and some example suppliers.

Arduino: The main site for the Arduino is <http://www.arduino.cc> This is the official site for all open-source information, and for software downloads.

A great secondary source for Arduino is <http://www.freeduino.org> which is a non-official community-supported site.

Arduino comes in many hardware variations. The main board that we're using is the Arduino Uno. It retails for around \$30. Some suppliers include:

Sparkfun: <http://www.sparkfun.com>

Radio Shack: <http://www.radioshack.com>

Adafruit: <http://adafruit.com>

Hacktronics: <http://www.hacktronics.com>

These are also great sources for all sorts of things related to embedded systems, physical computing, "making," and hardware hacking.

Switches and sensors: The previously mentioned sources also have loads of switches and sensors of all sorts. Many of the sensors that we didn't discuss in this workshop are also "resistive sensors" and can be used in exactly the same way as we used pots and light sensors.

Other sources for small electronic parts like this include:

Jameco: <http://www.mouser.com>

Digikey: <http://www.digikey.com>

Mouser: <http://www.mouser.com>

Of these suppliers, Jameco is more for the hobbyist, and Digikey and Mouser are more for the professional. They all sometimes have great prices, especially in quantity, but you sometimes really have to know what you're looking for.

As an example – an individual light sensor at Sparkfun or Jameco runs between \$1.25 and \$2.00. But, with a little more searching, Jameco has a bag of 50 assorted light sensors for \$15. Look for Jameco's bags of assorted switches, assorted LEDs, etc. They can be a great way to quickly stock a set of kits.

Servos: These are the type of servos used in radio controlled model airplanes and model cars, and for small robotics projects. So, any hobby shop will have them. Radio Shack, Sparkfun, etc. also carry them, but you might find a better deal at a hobby shop or a robotics hobbyist shop. Some examples:

Hobby Partz: <http://www.hobbypartz.com>

Trossen Robotics: <http://www.trossenrobotics.com>

Pololu: <http://www.pololu.com>

Electronics surplus resellers: Sometimes you can find great deals on surplus parts at a surplus reseller. Here you really do need to know exactly what you're getting, and be willing to take some chances, but you can find some real bargains if you look around. Some examples:

Electronics Goldmine: <http://www.goldmine-elec.com/>

Marlin P. Jones: <http://www.mpja.com/>

Alltronics: <http://www.alltronics.com/>

Electronics Surplus: <http://www.electronicssurplus.com/>

Lady Ada at Adafruit also has some great suggestions:

<http://www.ladyada.net/library/procure/hobbyist.html>