Embedded Systems and Kinetic Art

CS5968: Erik Brunvand
School of Computing

Art4455: Paul Stout
Department of Art and Art History

Logistics

- Class meets M-W from 11:50-2:50
- We'll start meeting in Sculpt 183
  - At some point we may also meet in MEB 3133 (Merrill Engineering Building) on the north side of campus
- Web page is www.eng.utah.edu/~cs5968
- TA is Josef Spjut
  - from School of Computing

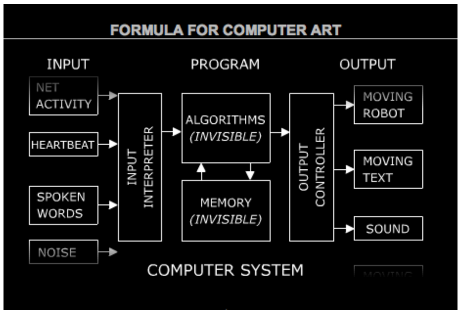
Kinetic Art

- **Art** that contains moving parts or depends on motion, sound, or light for its effect.
  - The kinetic aspect is often regulated using microcontrollers connected to motors, actuators, transducers, and sensors that enable the sculpture to move and react to its environment.

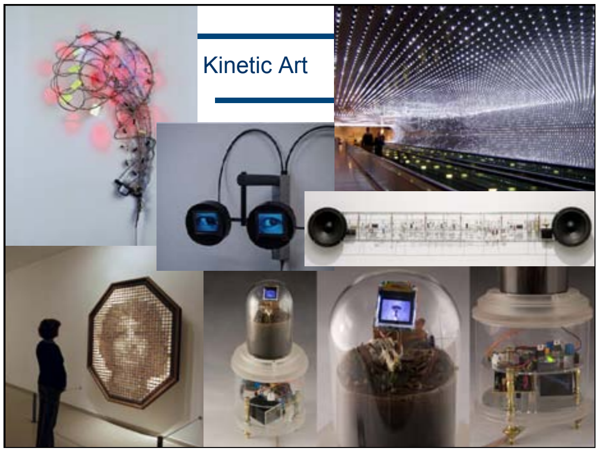
Embedded Systems

- A special-purpose computer system (microcontroller) designed to perform one or a few dedicated functions, often reacting to environmental sensors.
  - It is embedded into a complete device including hardware and mechanical parts rather than being a separate computer system.

Jim Campbell's Algorithm



Kinetic Art



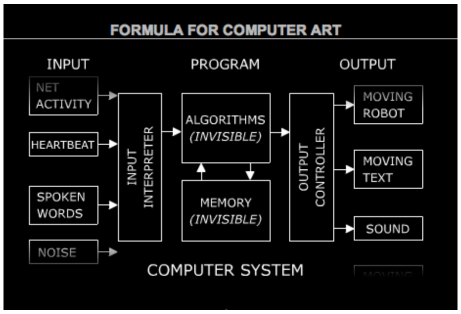
This Class

- Enabling engineers and artists to collaborate and make some interesting kinetic art
  - Artists and engineers to work in interdisciplinary teams
  - This will be a cross between an engineering class (embedded system design and programming) and an art studio class (designing and building the sculptures) with all students participating fully in both areas.

How will it Work?

- Good question! It's an ongoing experiment from both sides...
  - Start with some background study
  - Some hand's-on labs with the microcontroller
    - Build a toolkit of input sensors, output transducers and computer code to interface with them
  - Teams will eventually design a project together
  - Class critiques, refinement, final build
  - Exhibit of the results in Spring

Jim Campbell's Algorithm



Output Transducers

- Motion
  - Motors - DC, Stepper
  - Servos
- Light
  - LED, bulbs, etc.
- Sound
  - Generated, recorded, physical, etc.

Input Sensors

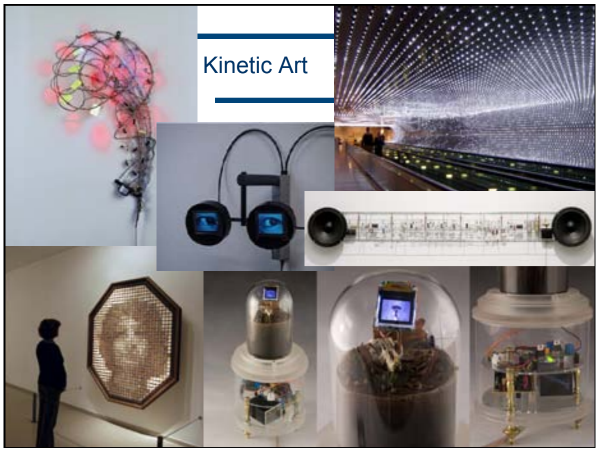
- Switches
- Resistive sensors
  - Get analog values based on sensing input
  - light, temperature, knobs, flex, etc
- Proximity/motion sensing
  - PIR, distance, etc.

Electronic Glue

- Power supplies
- Transistors
  - used as electronic switches for medium power devices
- Relays
  - used as electronic switches for high power devices
- resistors, capacitors, wires, etc.

Complete Art Piece

- Kinetic concept in a well-conceived and constructed artifact
  - Traditional 3d materials
  - Wood, metal, plastic, wiring, and other structural materials
  - Unattended functioning (i.e. in gallery)
  - Consider maintenance and support issues too…



Kinetic Art

Microcontroller

- The “brains” that coordinates the kinetics
  - Small computers
  - Typically with special support for sensors and actuators
    - Analog-digital converters on inputs
    - pulse-width modulation on outputs
- We’ll use one called Arduino

What is Arduino?

The word “Arduino” can mean 3 things

A physical piece of hardware

A programming environment

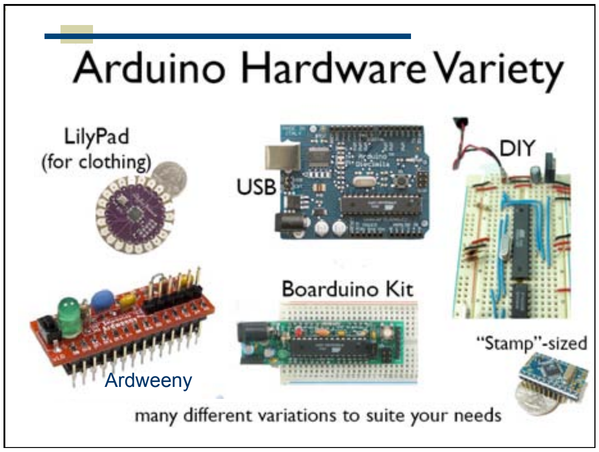
A community & philosophy

Arduino Community

- Open source physical computing platform
  - “open source” hardware
  - open source software environment
  - physical computing means sensing and controlling the physical world
- Community
  - Examples wiki (the “playground”)
  - Forums with helpful people

Arduino Hardware

- Similar to Basic Stamp (if you know of it)
  - but cheaper, faster, & open
- Uses AVR ATmega328p microcontroller chip
  - chip was designed to be used with C language



Arduino Hardware Variety

LilyPad (for clothing)

USB

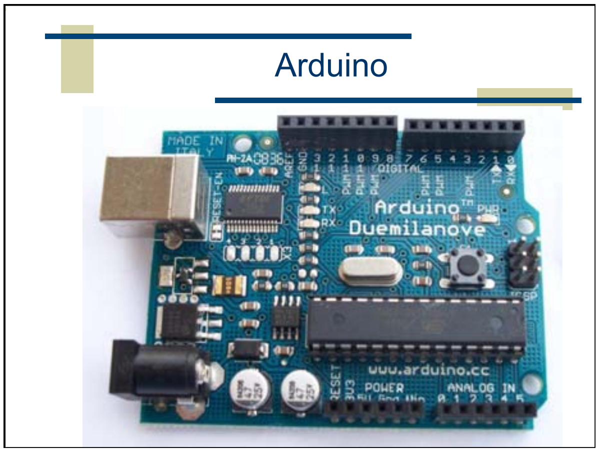
DIY

Boarduino Kit

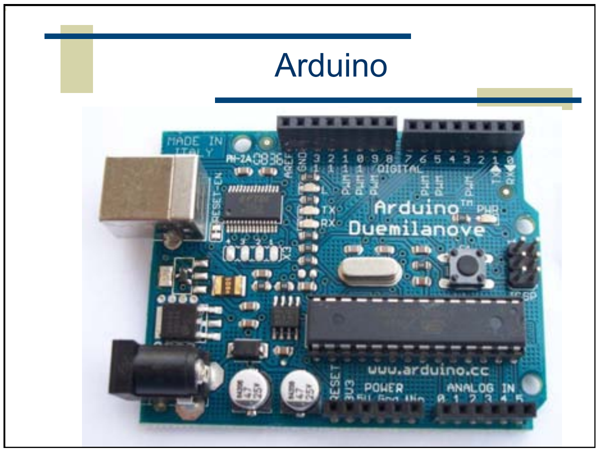
"Stamp"-sized

Ardweeny

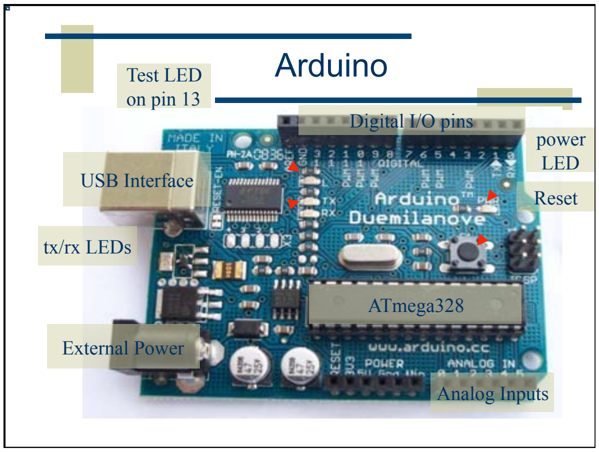
many different variations to suite your needs



Arduino



Arduino



Arduino

Test LED on pin 13

Digital I/O pins

power LED

Reset

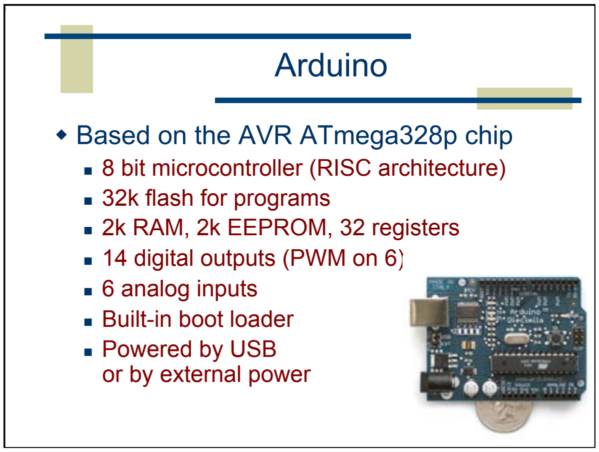
USB Interface

tx/rx LEDs

ATmega328

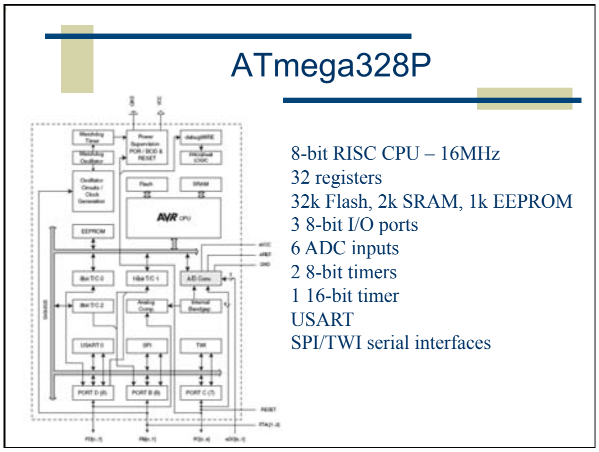
External Power

Analog Inputs



Arduino

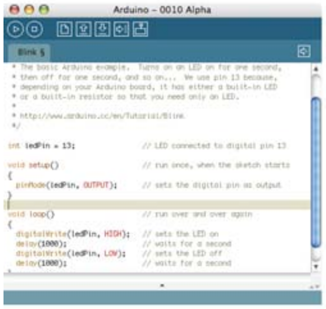
- Based on the AVR ATmega328p chip
  - 8 bit microcontroller (RISC architecture)
  - 32k flash for programs
  - 2k RAM, 2k EEPROM, 32 registers
  - 14 digital outputs (PWM on 6)
  - 6 analog inputs
  - Built-in boot loader
  - Powered by USB or by external power



ATmega328P

8-bit RISC CPU – 16MHz
32 registers
32k Flash, 2k SRAM, 1k EEPROM
3 8-bit I/O ports
6 ADC inputs
2 8-bit timers
1 16-bit timer
USART
SPI/TWI serial interfaces

Arduino Software

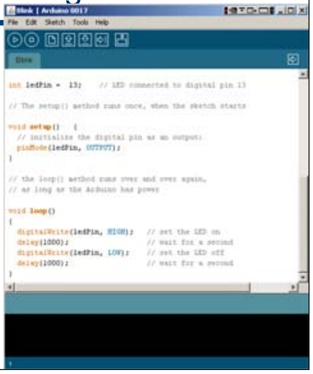


```
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

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

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  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
```

- Like a text editor
- View/write/edit sketches
- But then you program them into hardware

Programming Arduino

- Open-source programming environment
- Arduino language is based on C
  - Actually, it *is* C/C++
  - Hiding under the hood is gcc-avr
  - But, the Ardiuino environment has lots of nice features to make programming less scary...



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

// The setup() method runs once, when the sketch starts

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

// the loop() method runs over and over again,
// as long as the Arduino has power

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

More Arduino Info?

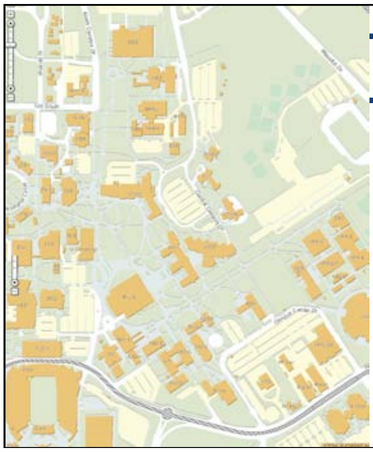
- www.arduino.cc/
  - Main Arduino project web site
- www.arduino.cc/playground/Main/HomePage
  - "playground" wiki with lots of users and examples
- www.freeduino.org/
  - "The world famous index of Arduino and Freeduino knowledge"
- www.eng.utah.edu/~cs5968
  - our class web site

Resources for this class

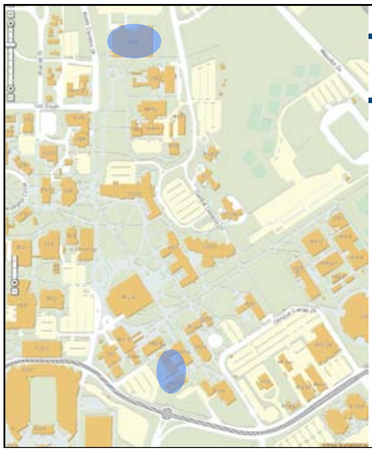
- We have some supplies for the class
  - Arduino boards
  - sensors of various different types
  - motors and servos
  - LEDs and LED controllers
- You should expect to have to buy a few more parts on your own to complete your project though...
  - We can use the electronics lab in the School of Computing, and wood and metal shop facilities in Art

Wedesday

- We'll do a hand's-on session with the Arduino boards
  - Bring a laptop if you have one
  - We'll write some very simple programs
  - Interface to some very simple sensors/LEDs
- Meet in Merrill Engineering Building
  - Room MEB 3133



Where?



Where?

MEB 3133 (DSL)

- Third Floor
- Main NS hallway on the West side of building
- Look for candy machines and metal stairs
- East into hallway
- Recessed door to 3133

Questions?