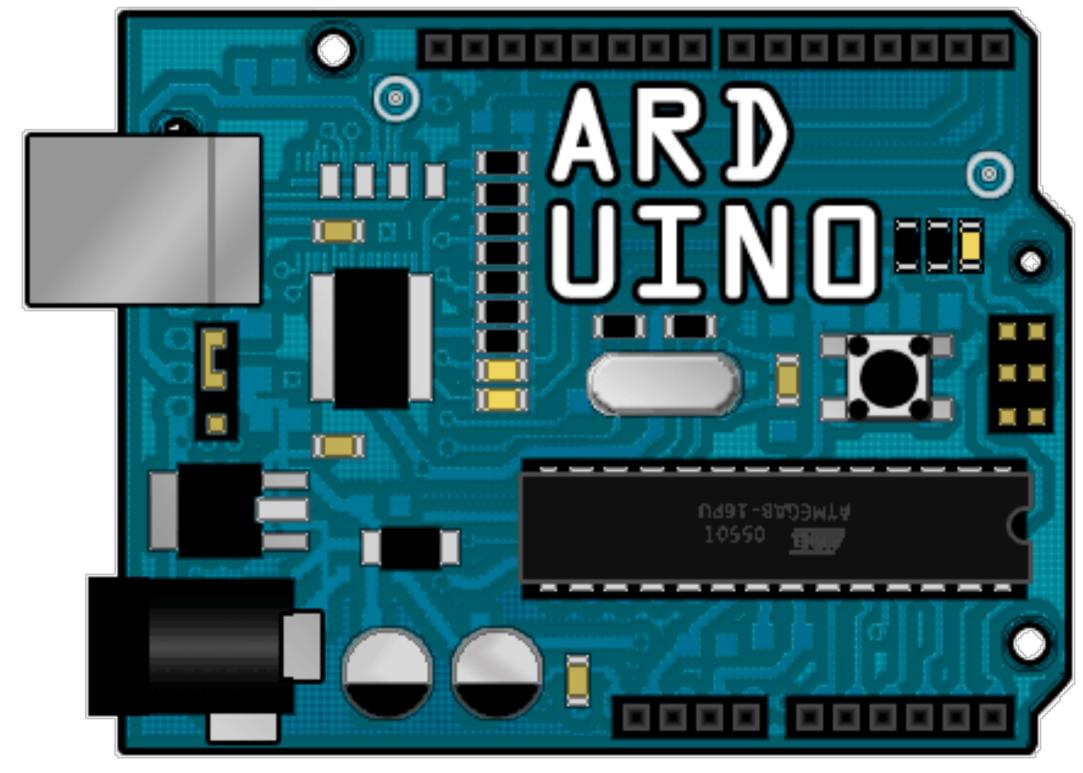


Introducing the Arduino

CS4062 - Eoin Brazil - Semester 2 - 2009





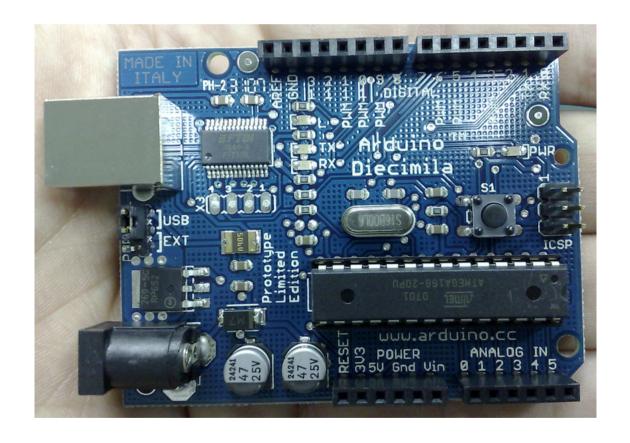


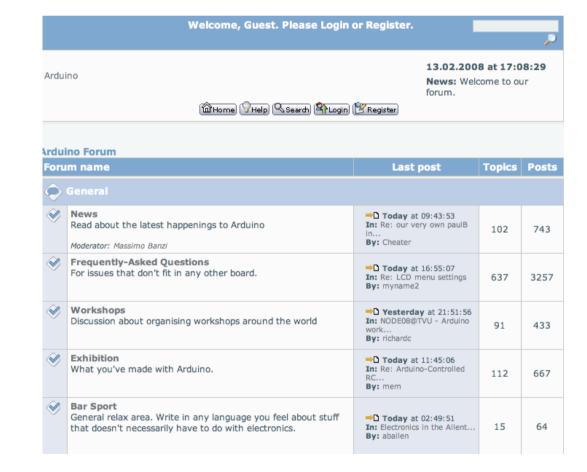
What is Arduino?

The hardware

The development environment

The community









Why Arduino?



artists & designers



x "opportunistic prototyping"



device hacking & reuse



💢 "open source hardware"



Computing Platform



open source



free to inspect & modify



community



wiki, forums, tutorials





The Nuts & Bolts

physical computing. er, what? ubiquitous computing, pervasive computing, ambient intelligence, calm computing, everyware, spimes, blogjects, smart objects...

tiny computer you can program

completely stand-alone, talks to other devices

☆ 'C'

😭 Flash

Processing

☆ PD

Max/MSP

Ruby

Python

PHP

Matlab

Squeak (Smalltalk)





"Physical Computing is about prototyping with electronics, turning sensors, actuators and microcontrollers into materials for designers and artists."

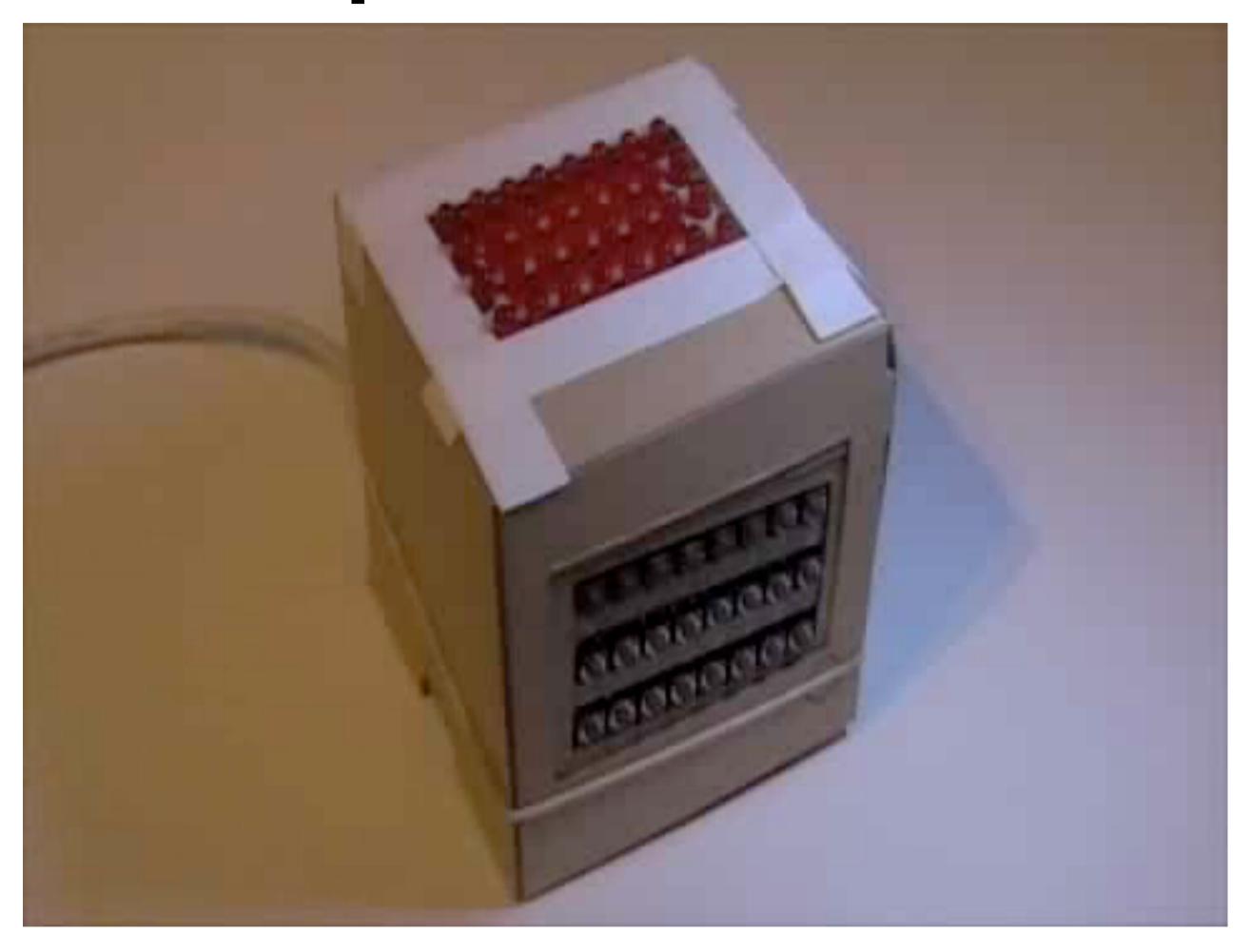
"It involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behaviour implemented as software running inside a microcontroller."

Massimo Banzi, Tinker.it & Arduino Co-Founder





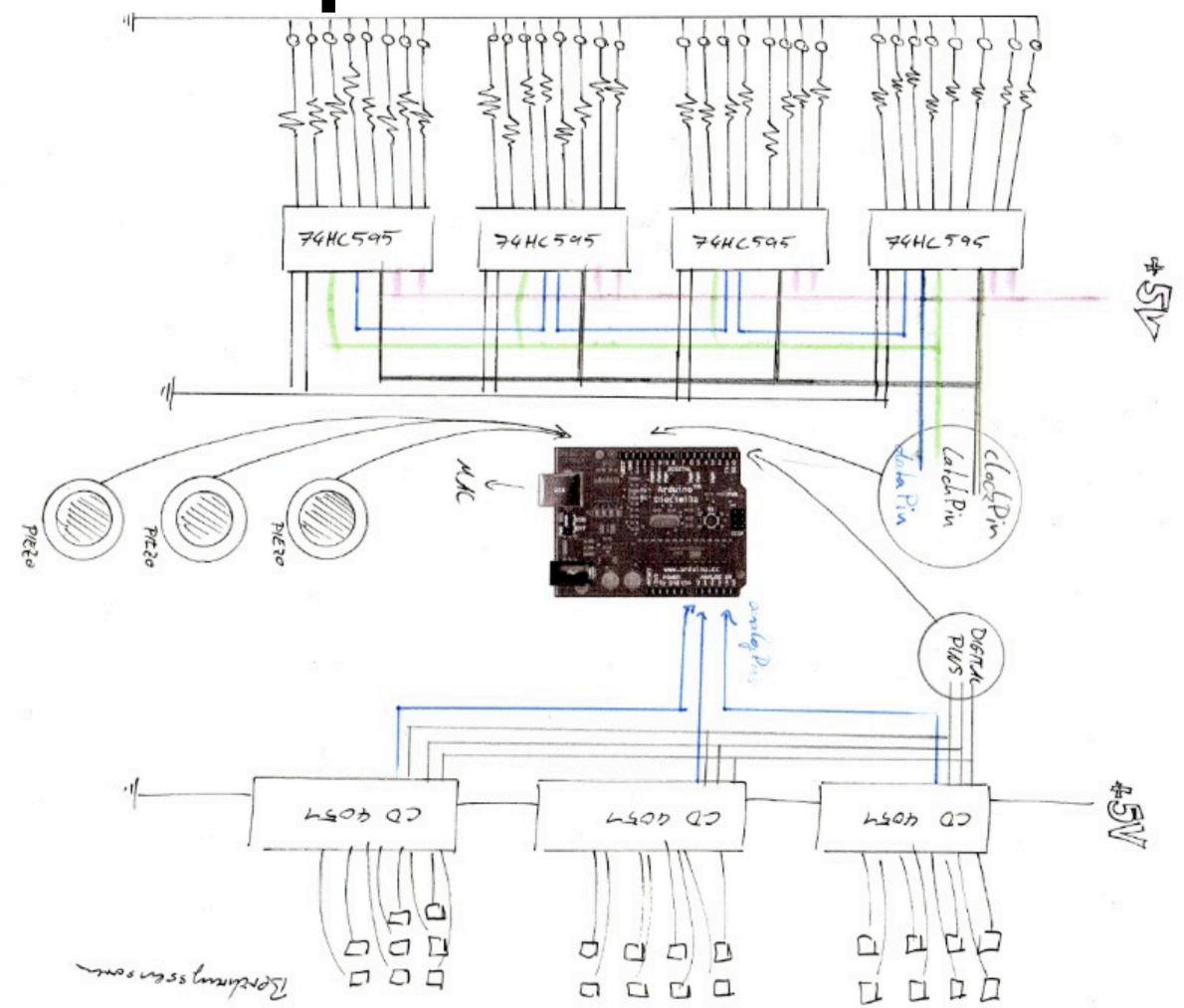
design centre







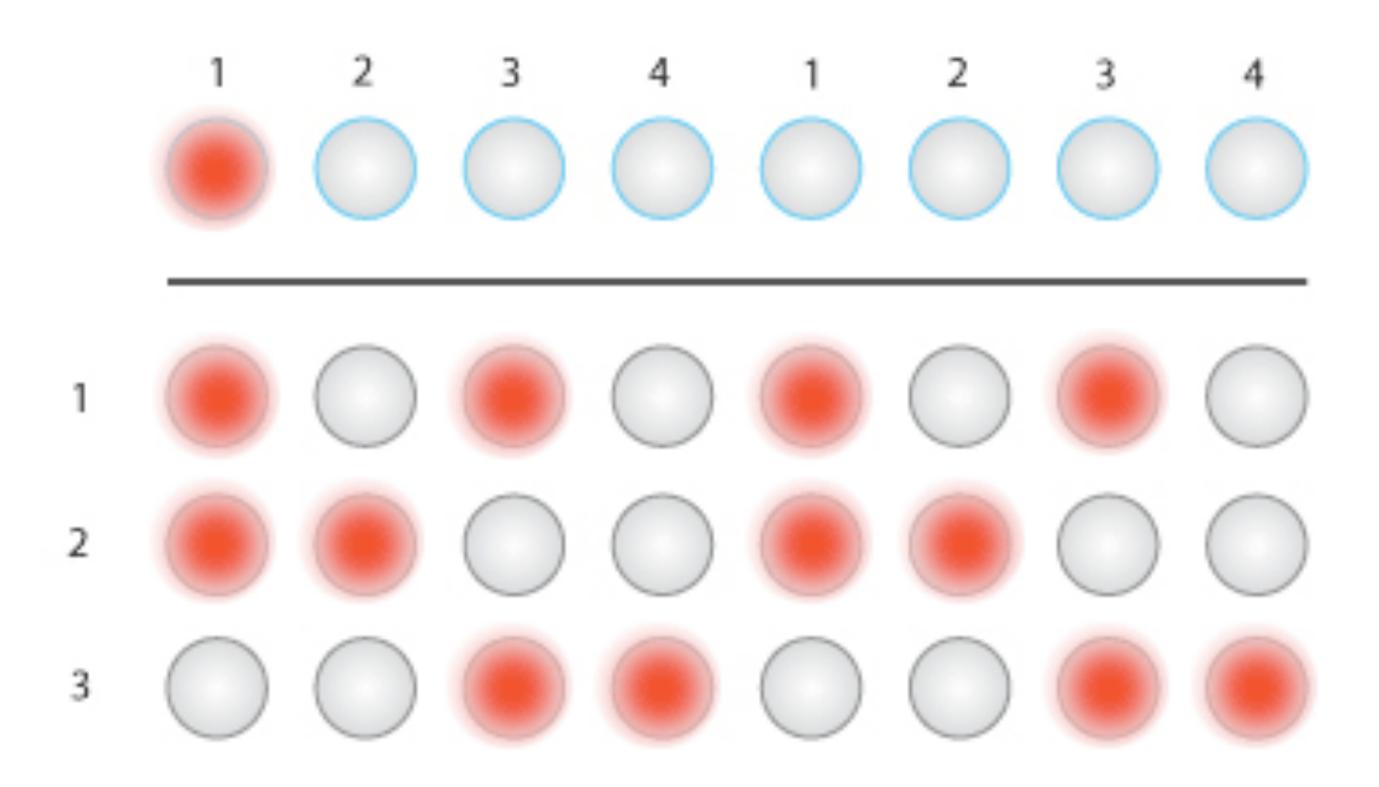
interacti design centre







design centre



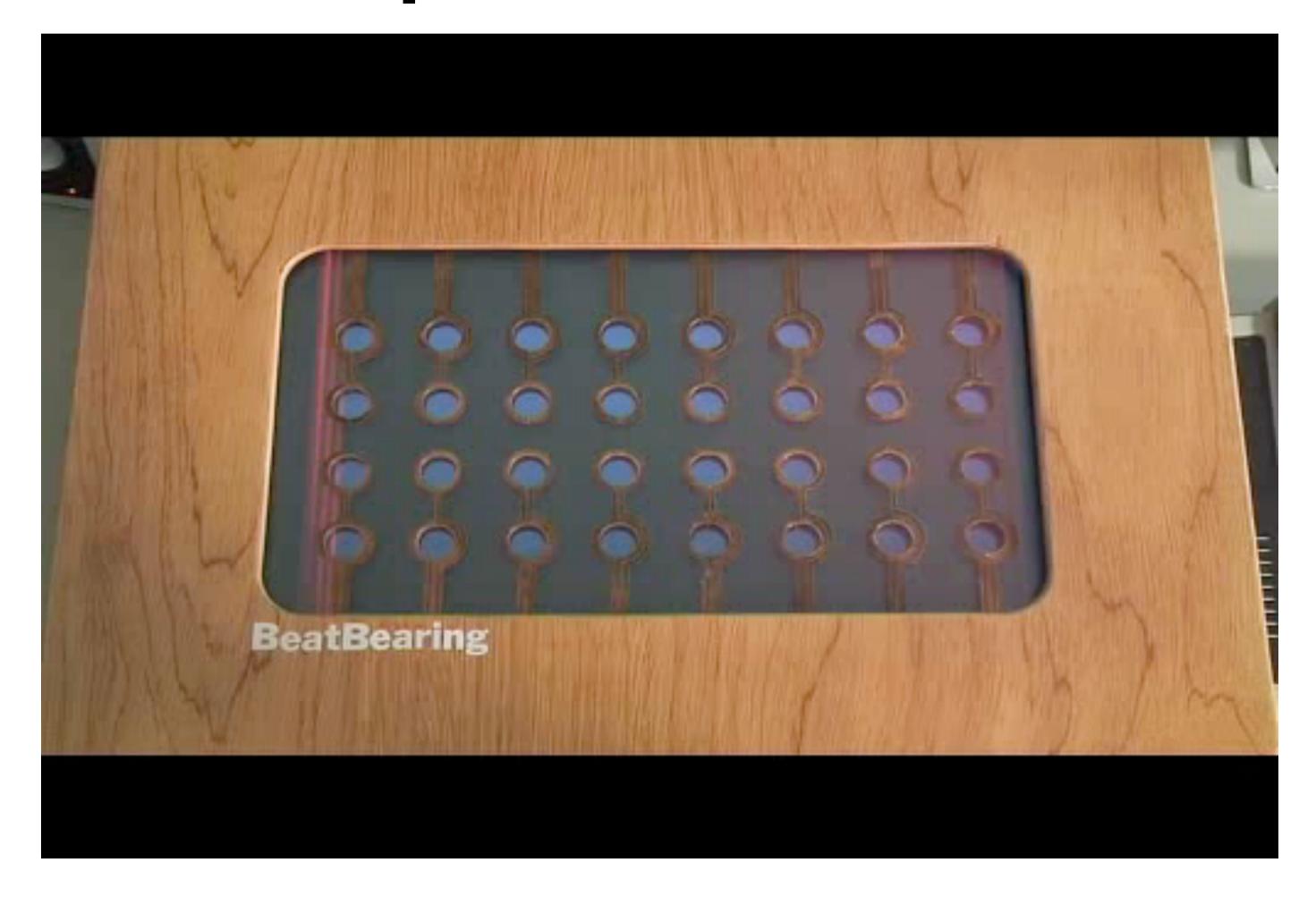












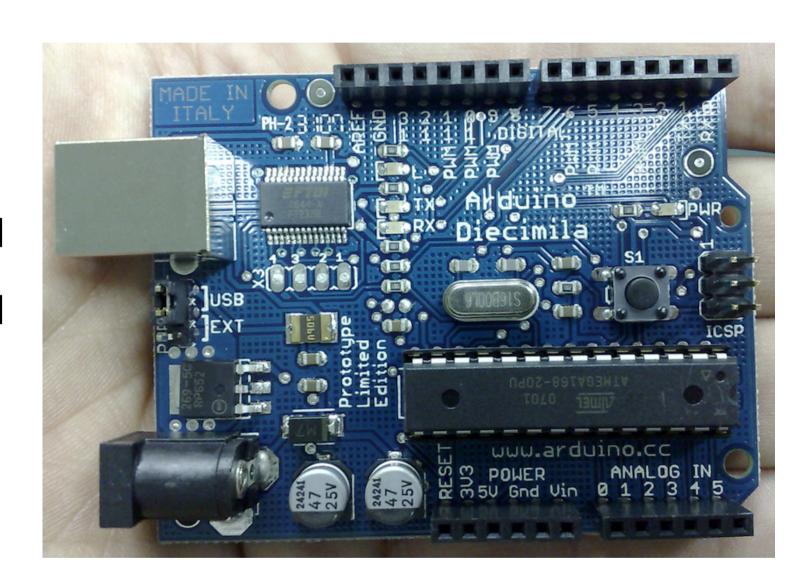




Arduino Capabilities











Arduino Capabilities



16 kBytes of Flash program memory



I kByte of RAM



X 16 MHz (Apple II: I MHz / Intel 286: 12.5 MHz / Intel Core 2:3 GHz)

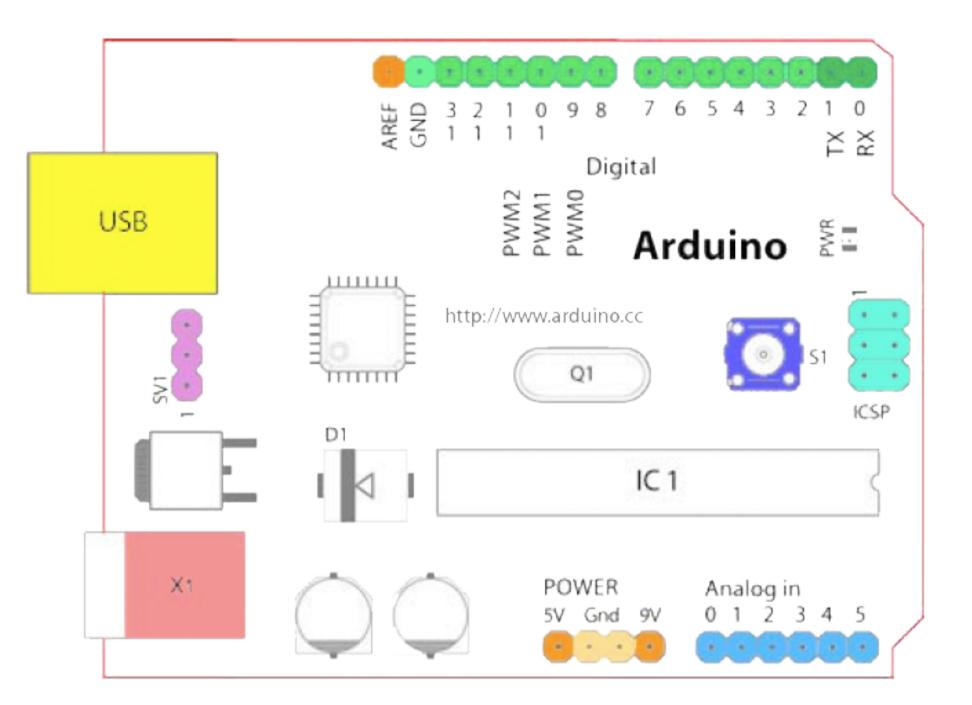


inputs and outputs

- 13 digital input/output pins
- 5 analog input pins
- 6 analog output pins (PWM only)







Layout of an Arduino

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out TX/RX (dark green)

These pins cannot be used for digital i/o (digitalRead and digitalWrite) if you are also using serial communication (e.g. Serial.begin).

- Reset Button SI (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) XI (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) SVI (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)



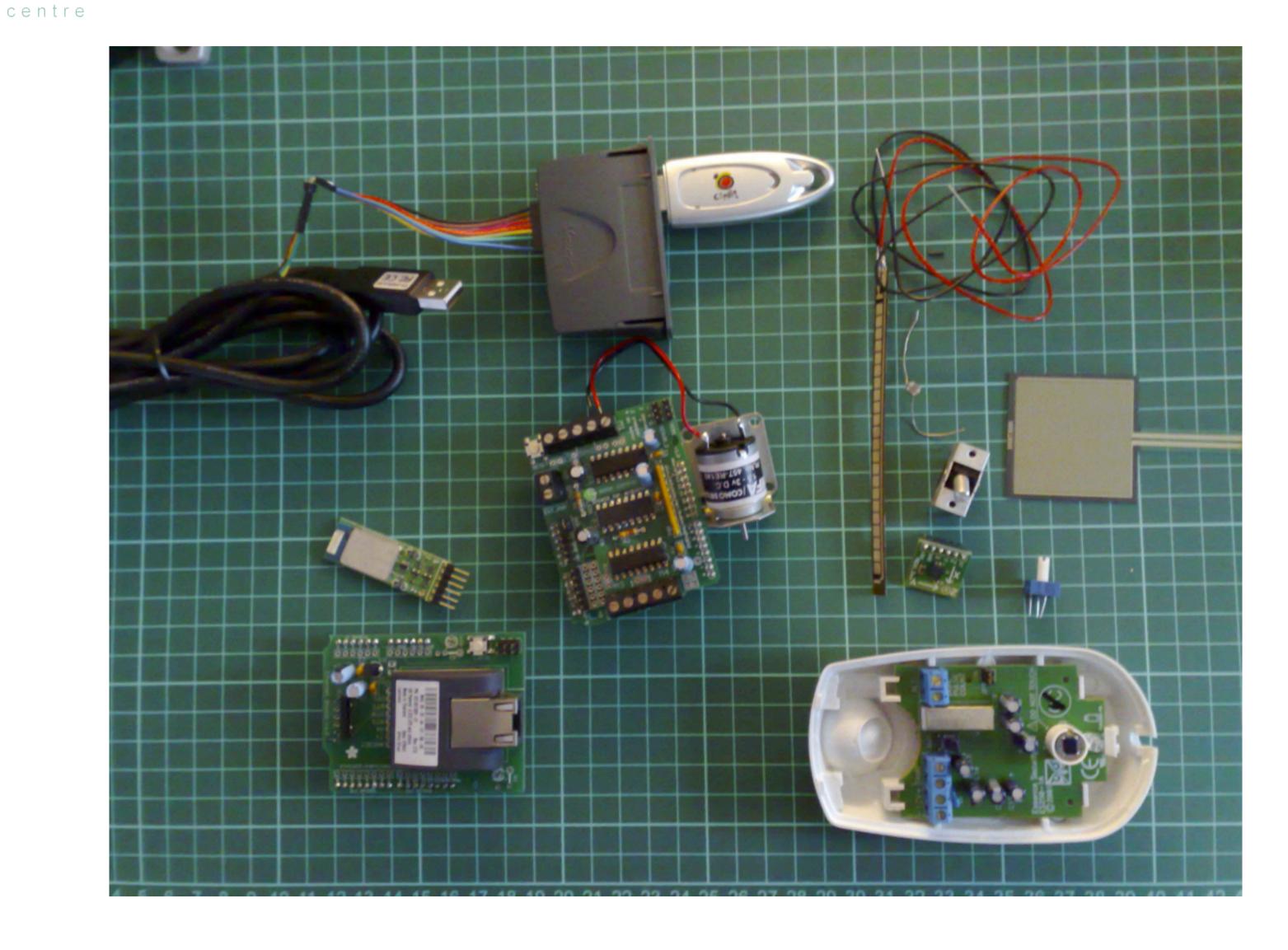


Arduino Glossary

- *****
 - ``sketch"- program that runs on the board
- `pin''- input or output connected to something, e.g. output to an LED, input from switch
- '`digital''- I (HIGH) or 0 (LOW) value (i.e. on/off)
- ``analog''- range (0-255 typically), e.g. LED brightness
- wiring drawing shows the interconnection (physical layout) of all devices and components
- schematic drawing shows underlying logic may not link back to physical layout

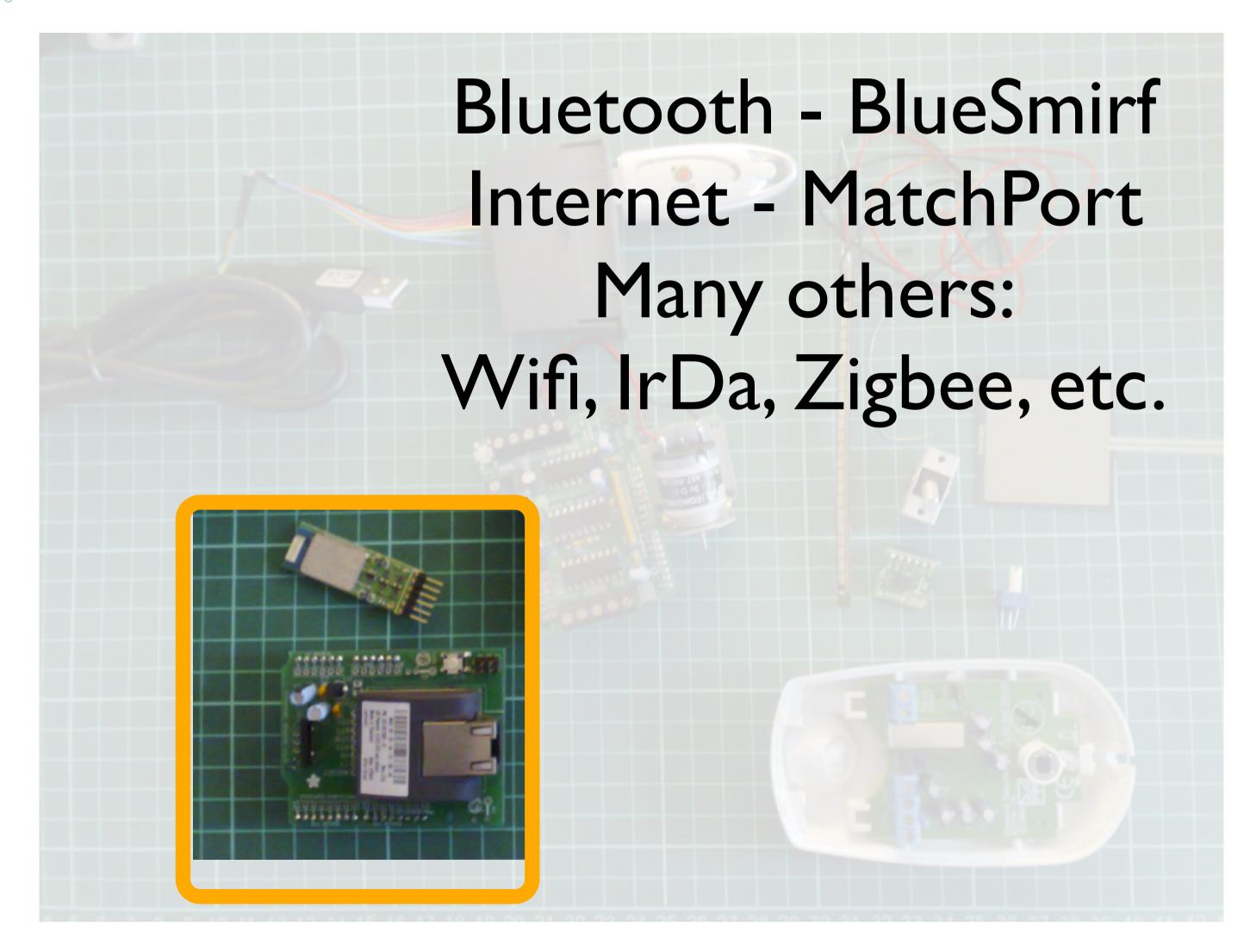






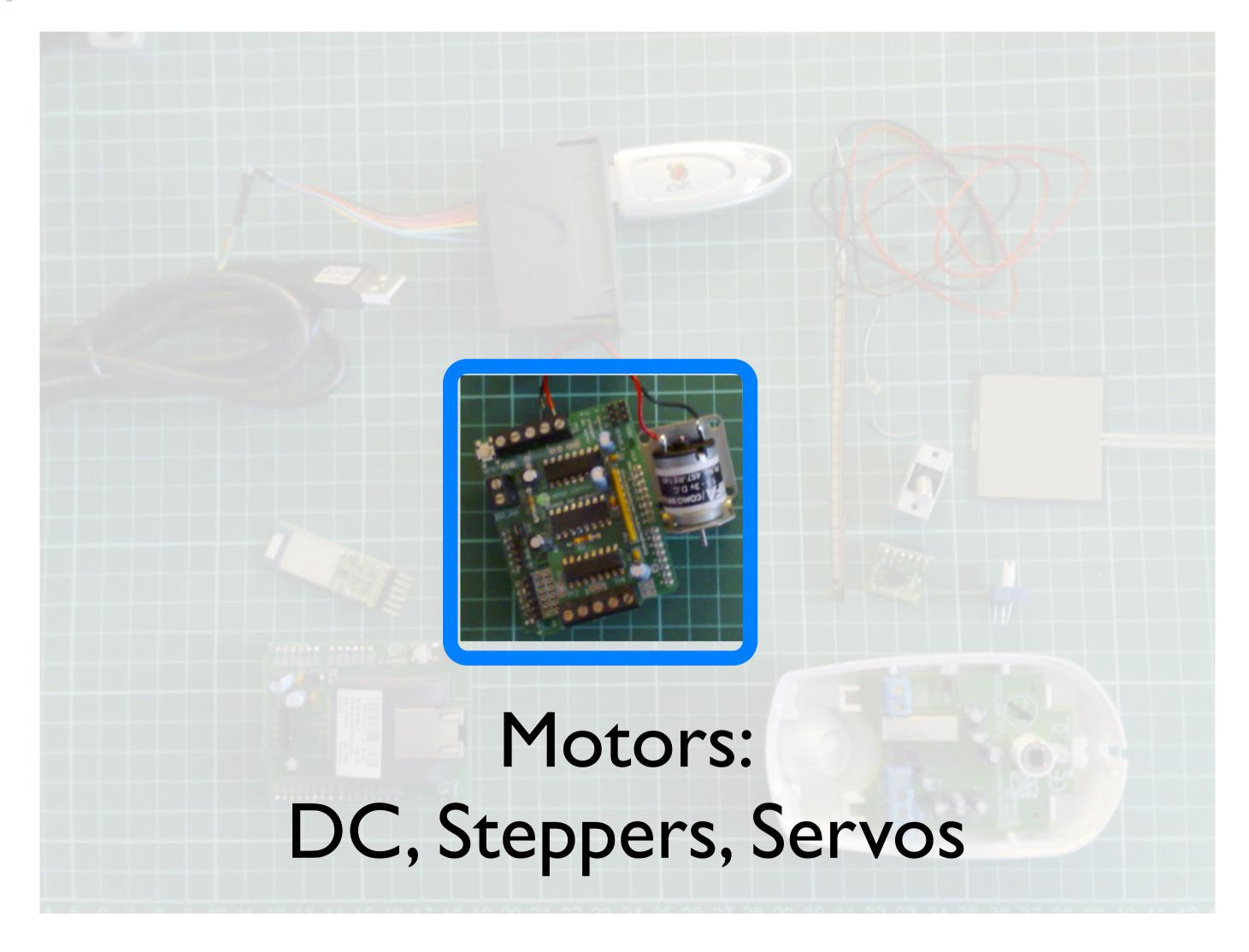






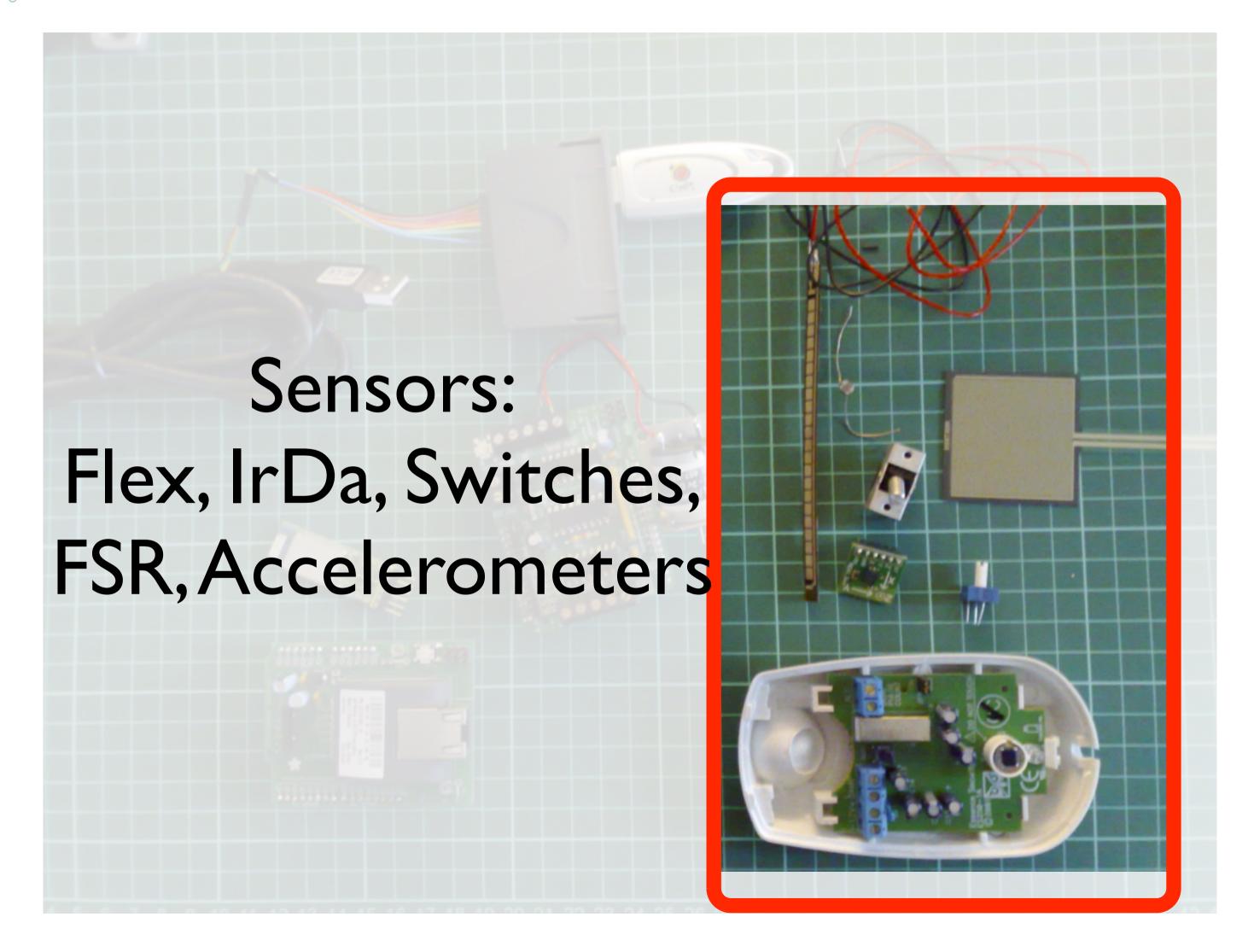






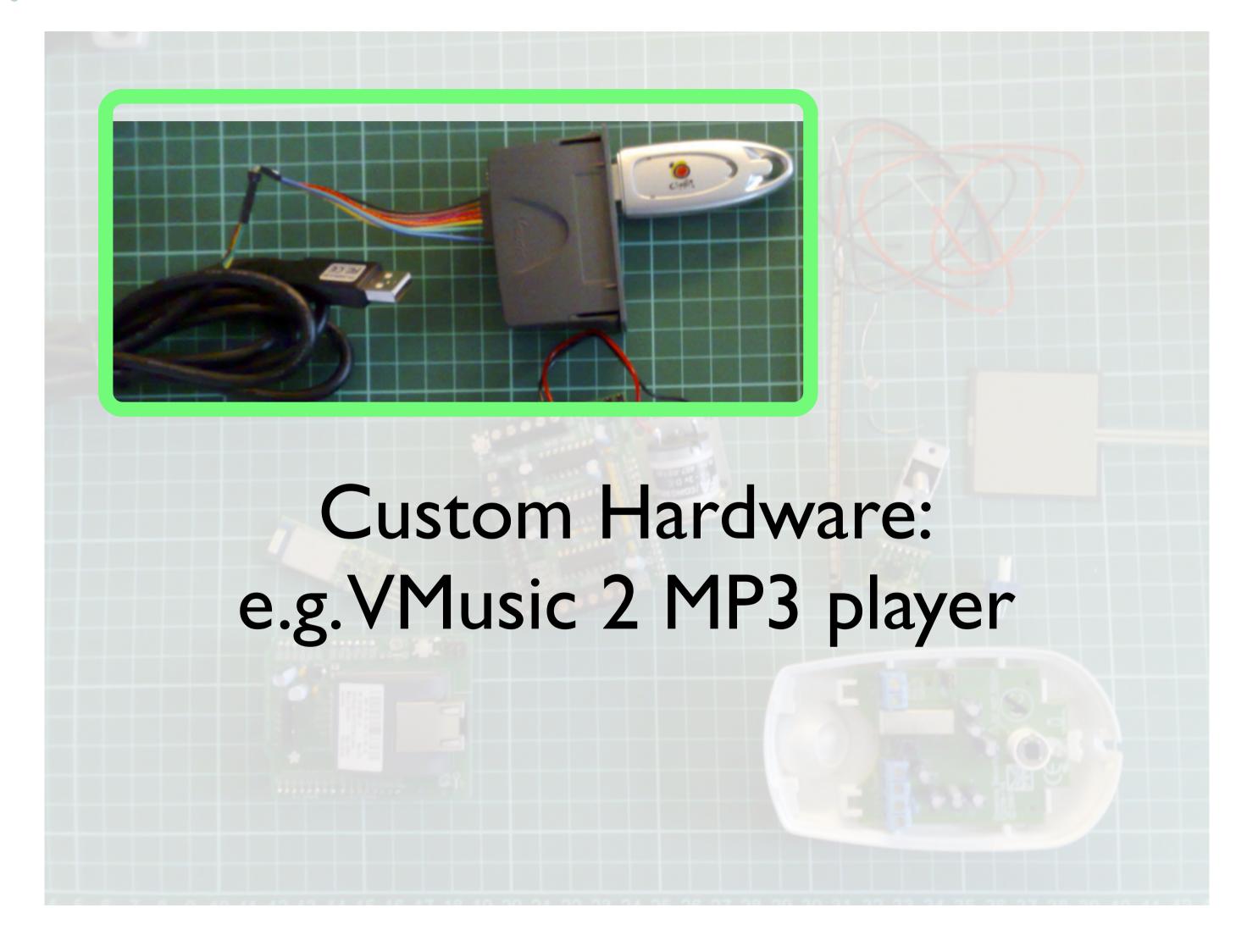










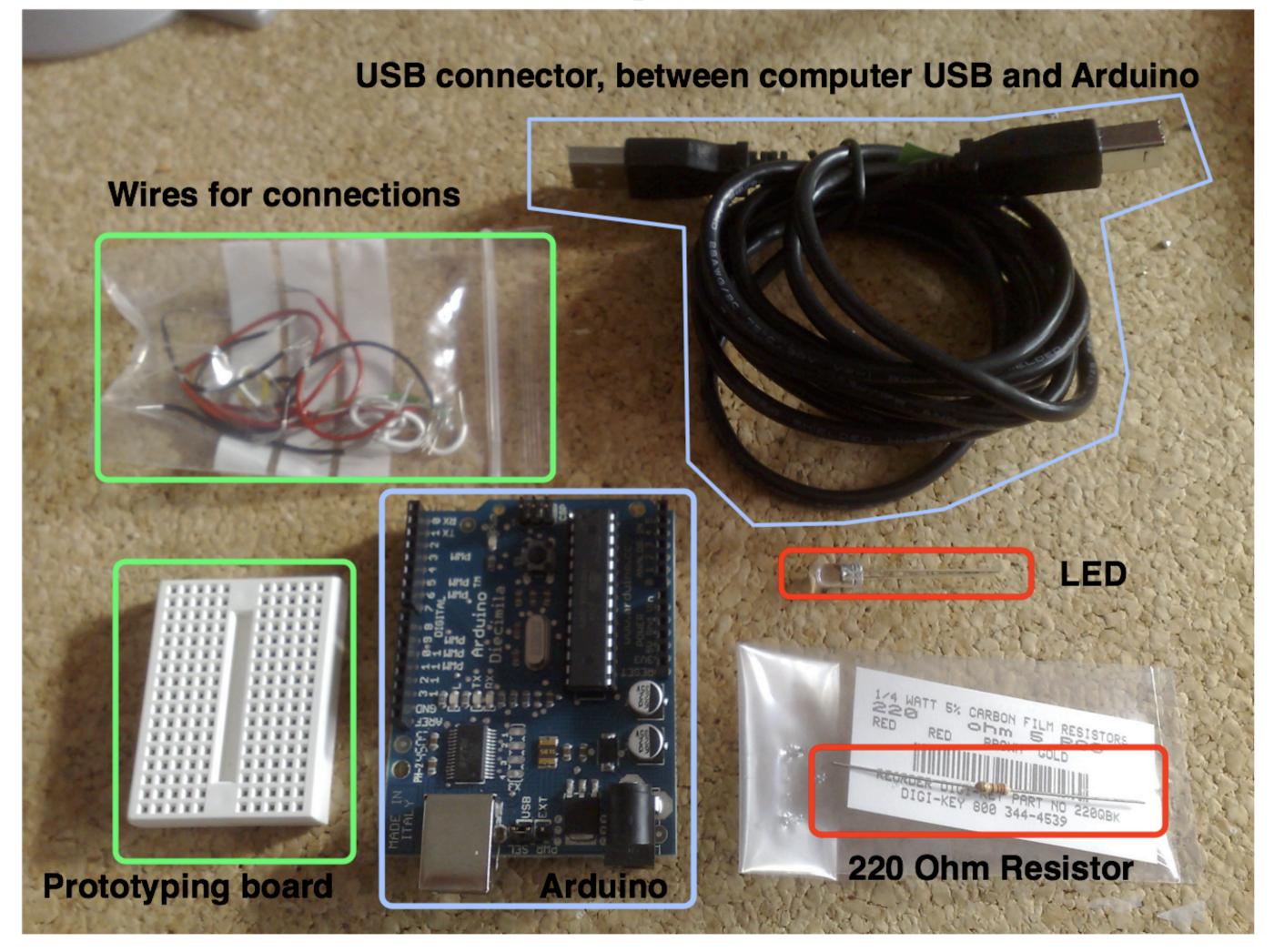






interaction design centre

Blinking a LED

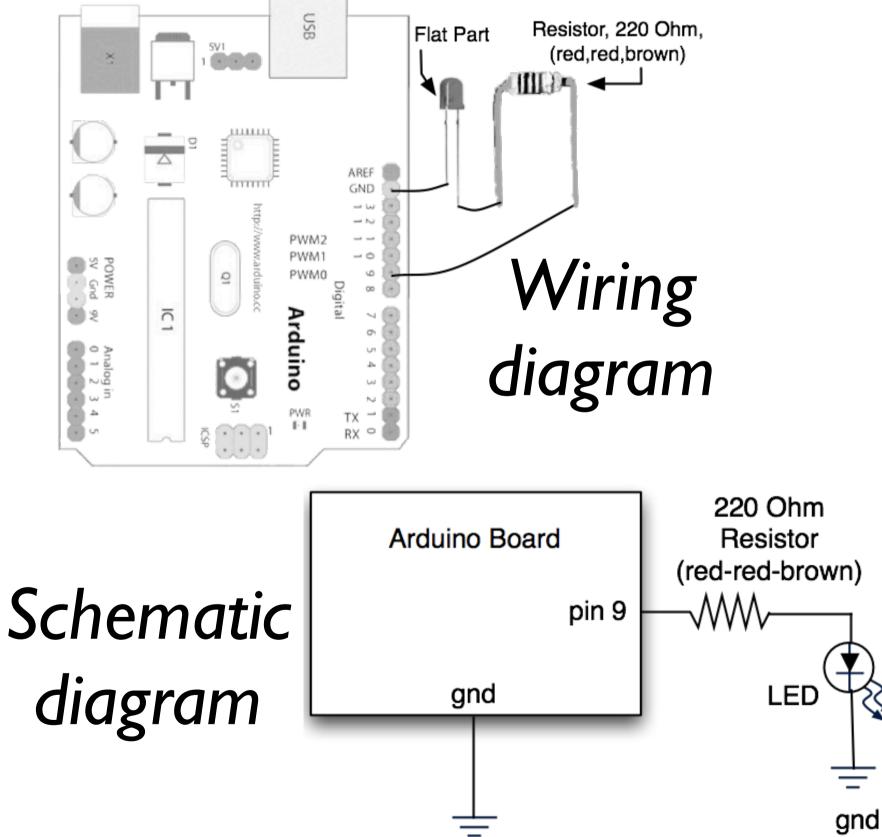


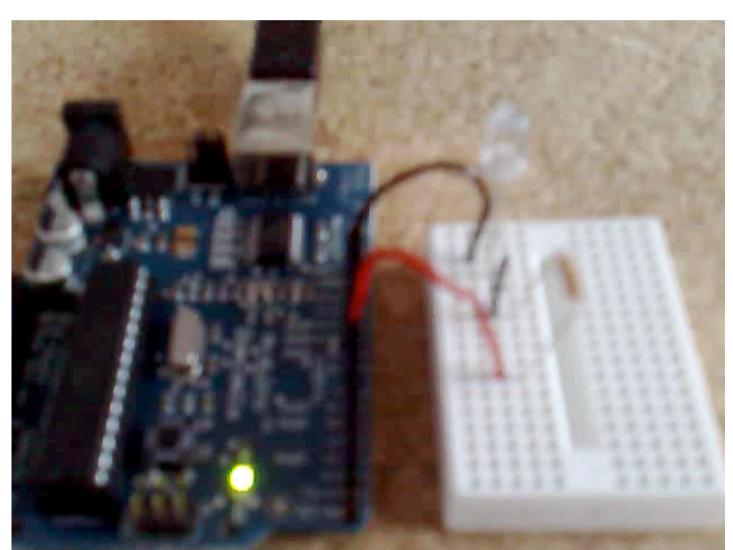




design centre

Diagrams



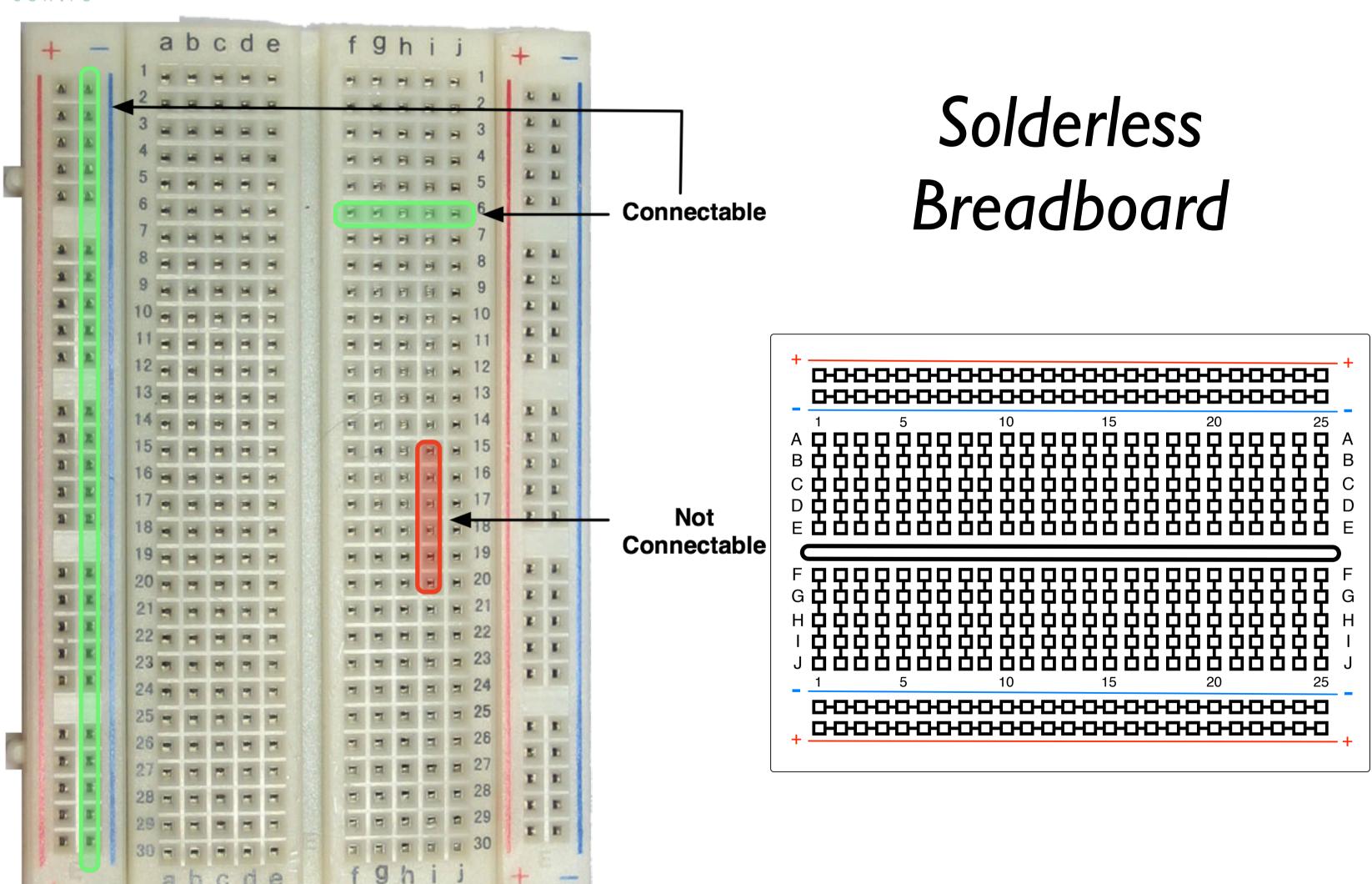


Wired circuit





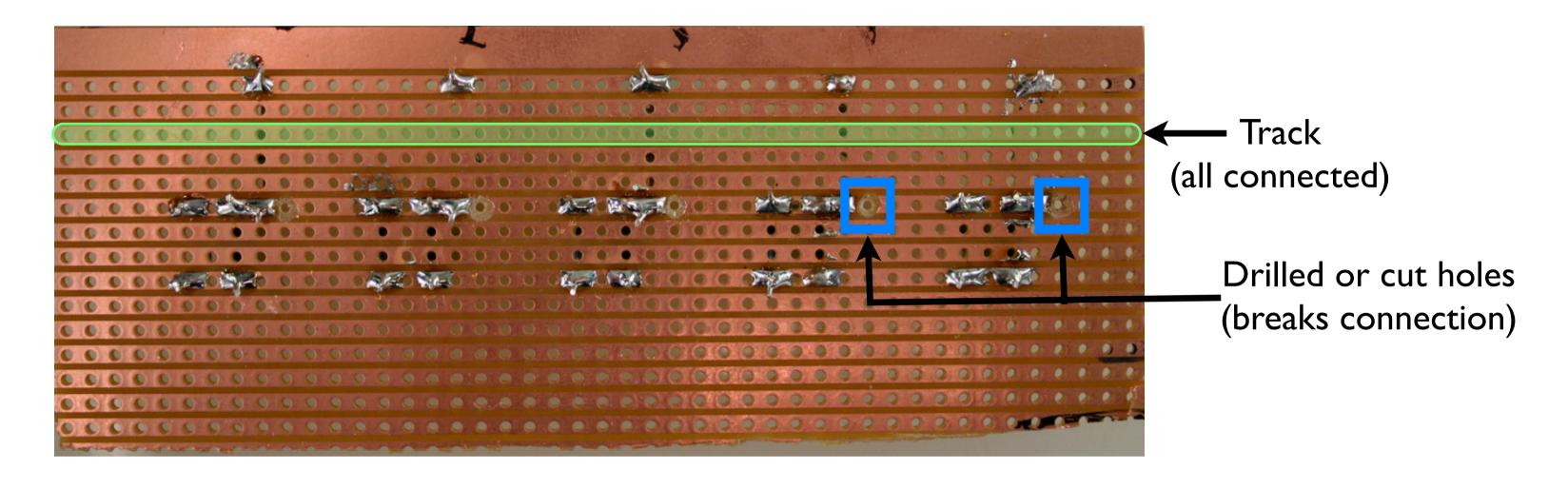
interaction
design
centre

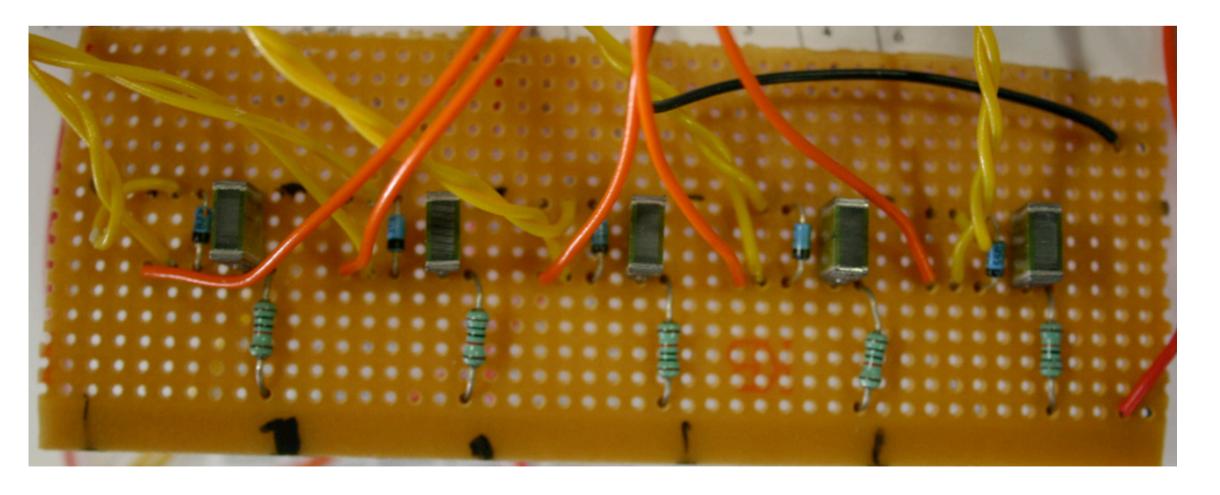






interaction design centre





Stripboard /
Perfboard
Example





ınteractıon design

centre

••••	•	•	•	•					•		•		•	•			•			•	•	•	•	lacktriangle
••••	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
••••	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
••••	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
••••	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
••••	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
• • • •		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
• • • •	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
••••	, ,	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
••••	$\overline{}$		_		_	_		_						_	_	_		_		_				

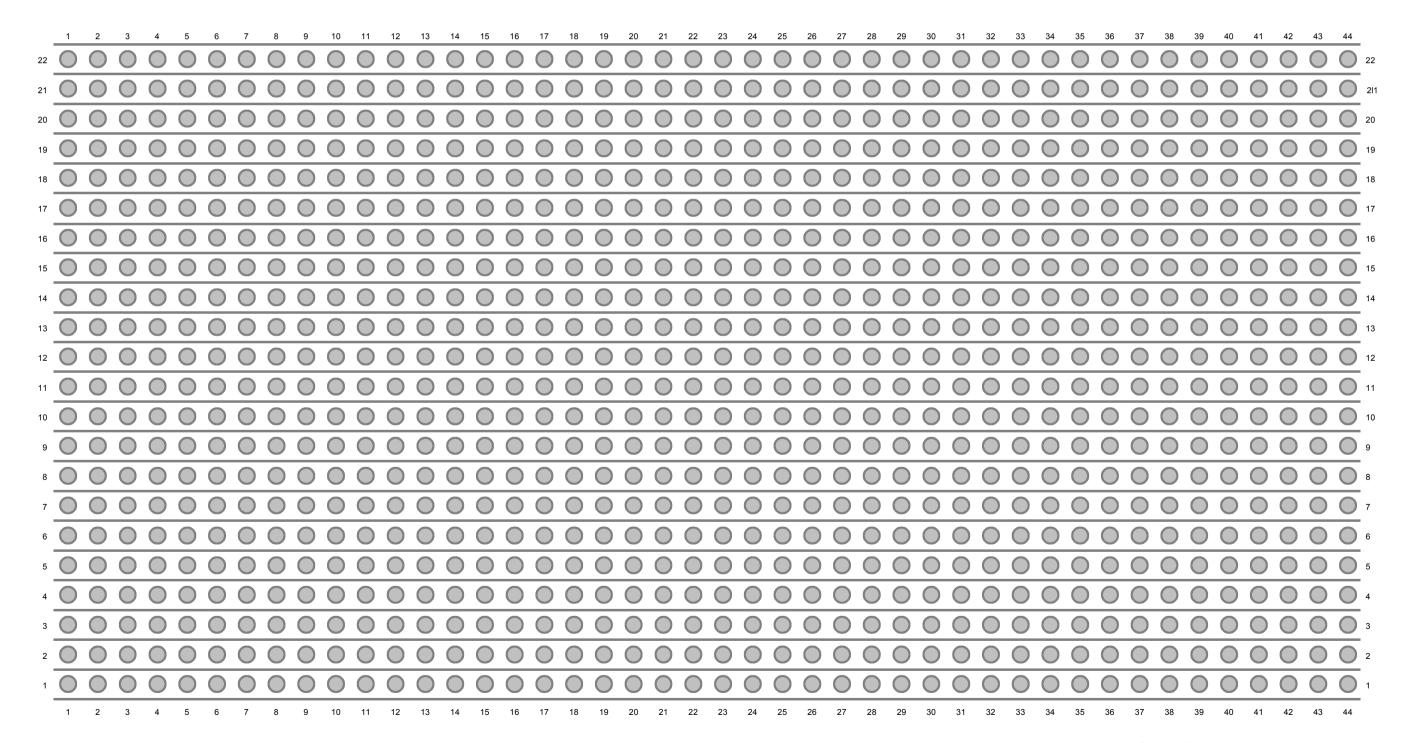
Tracks run this way

Tips:

- Mark out the Vs and 0V power lines first, then place the ICs.
- Remember to cut the track between the pins of an IC. Mark the cuts on the diagram with an X.
- Try to make resistors and axial capacitors lay flat on the stripboard. Resistors usually require a gap of 4 holes, capacitors a gap of 8 holes.
- Use the actual size grid on the left to check component spacing.
- Number the pins of the ICs as shown.

ELECTRONICS MECCANO

Strippo	pard	Layout	Planning	Snee
Project:				
Designed by:				
Version:				
Date:				
Notes:				



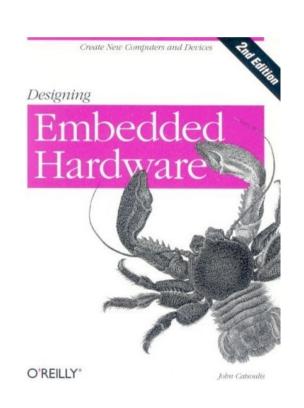
Version 3: 3/02 © 2002 Electronics in Meccano www.eleinmec.com



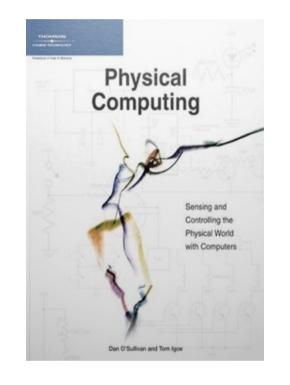




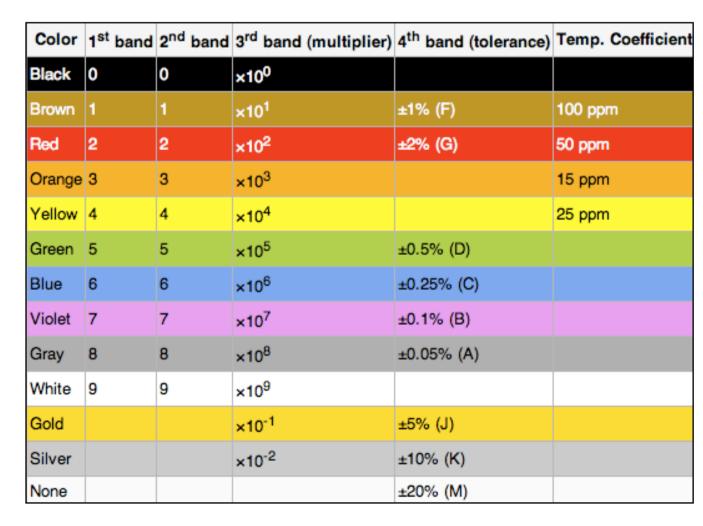
Books & first time prototyping advice

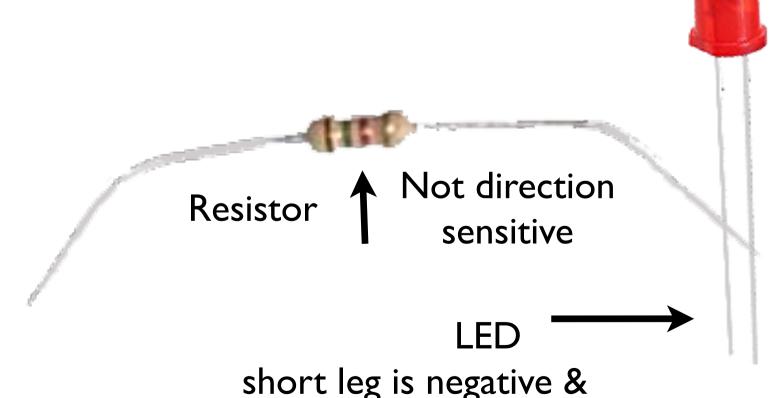






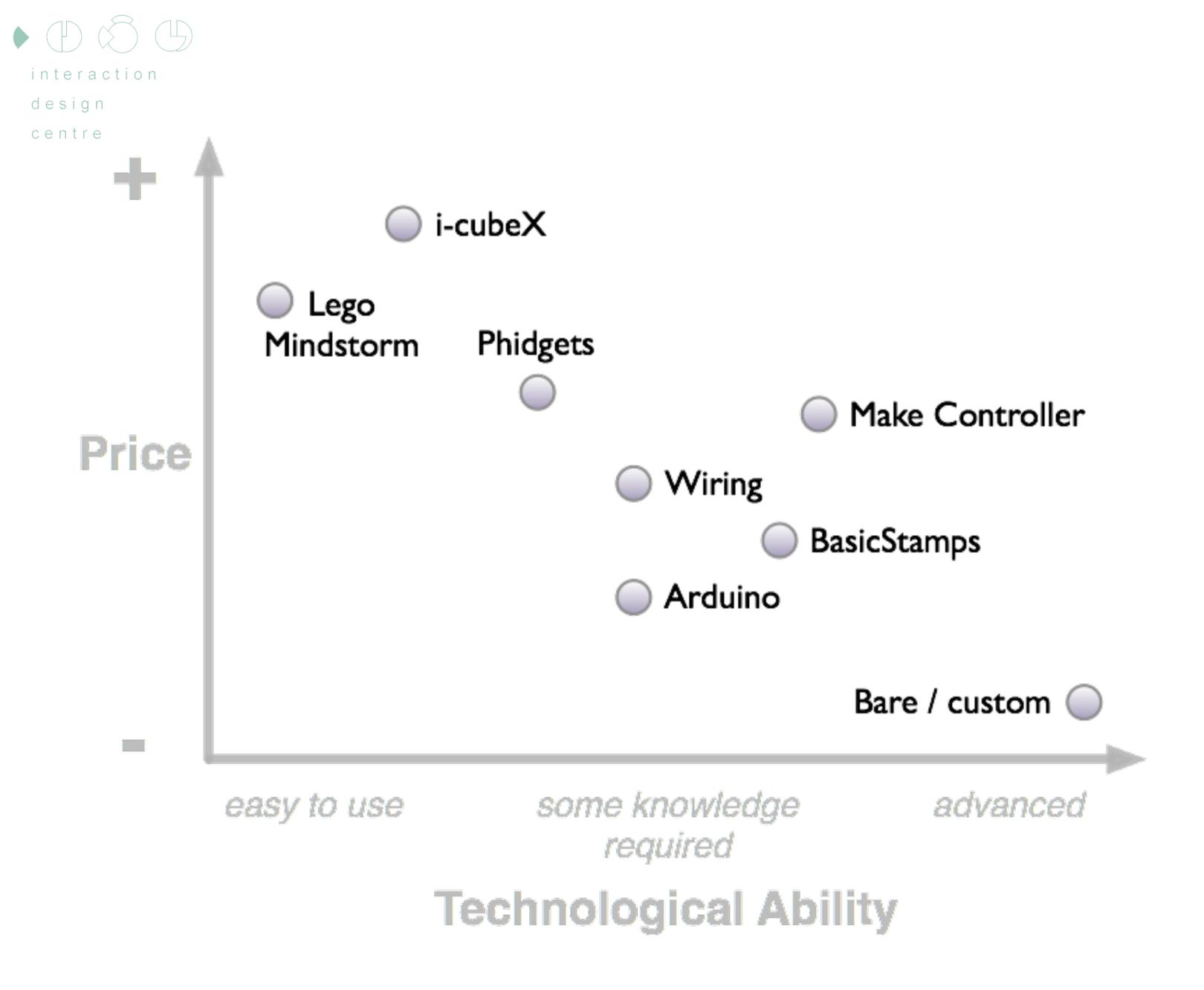
Blogs, YouTube, and Google



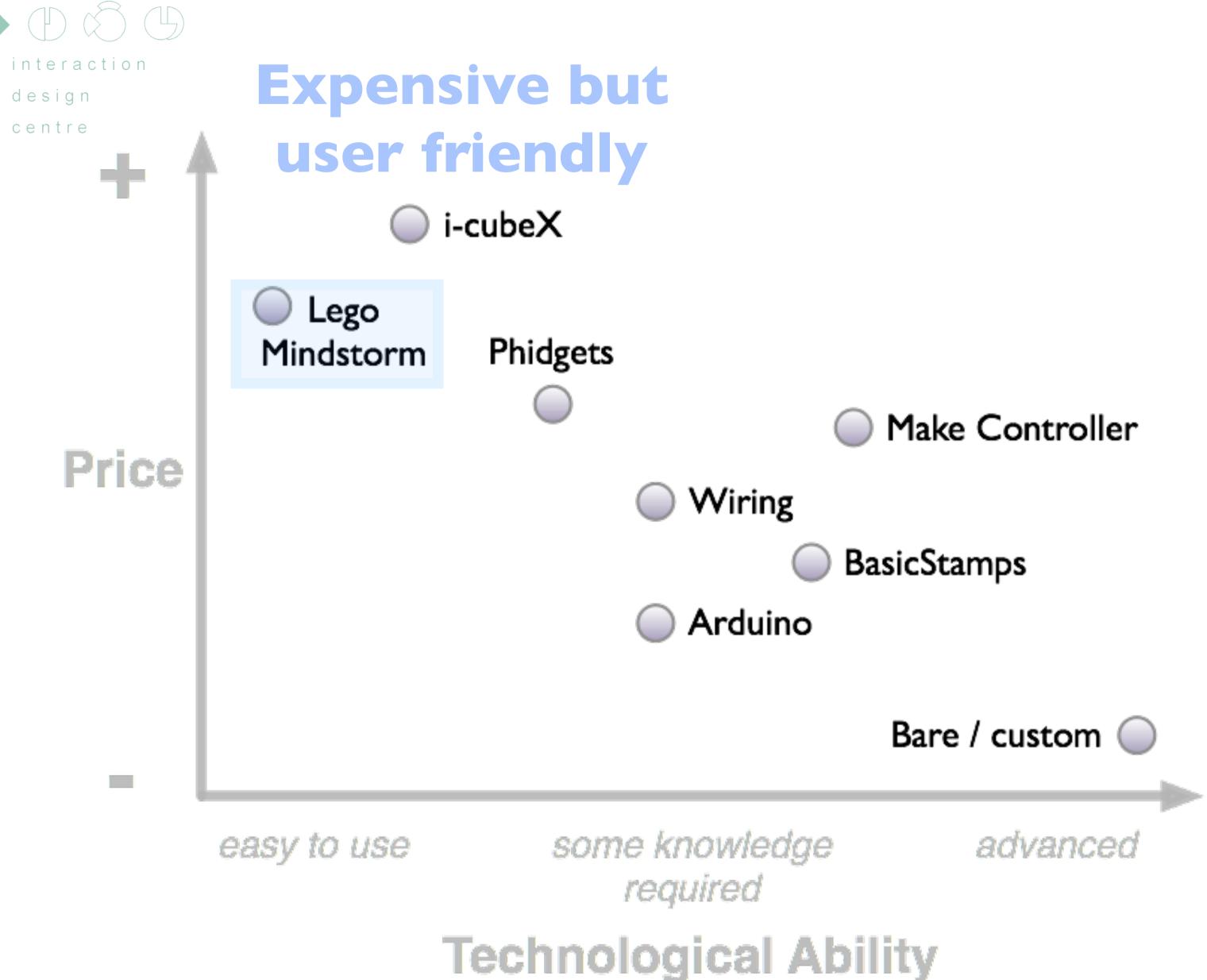


goes to the ground

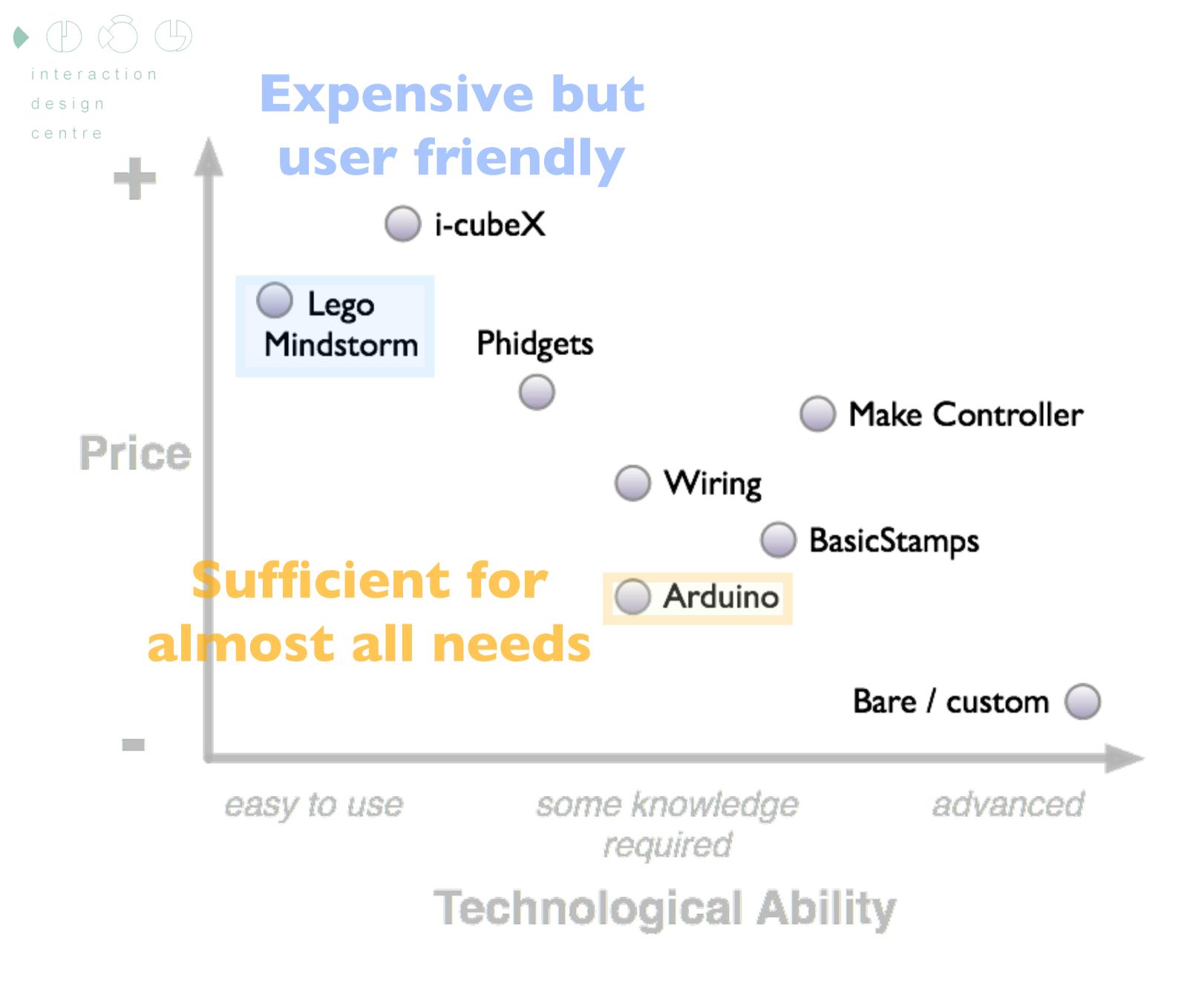




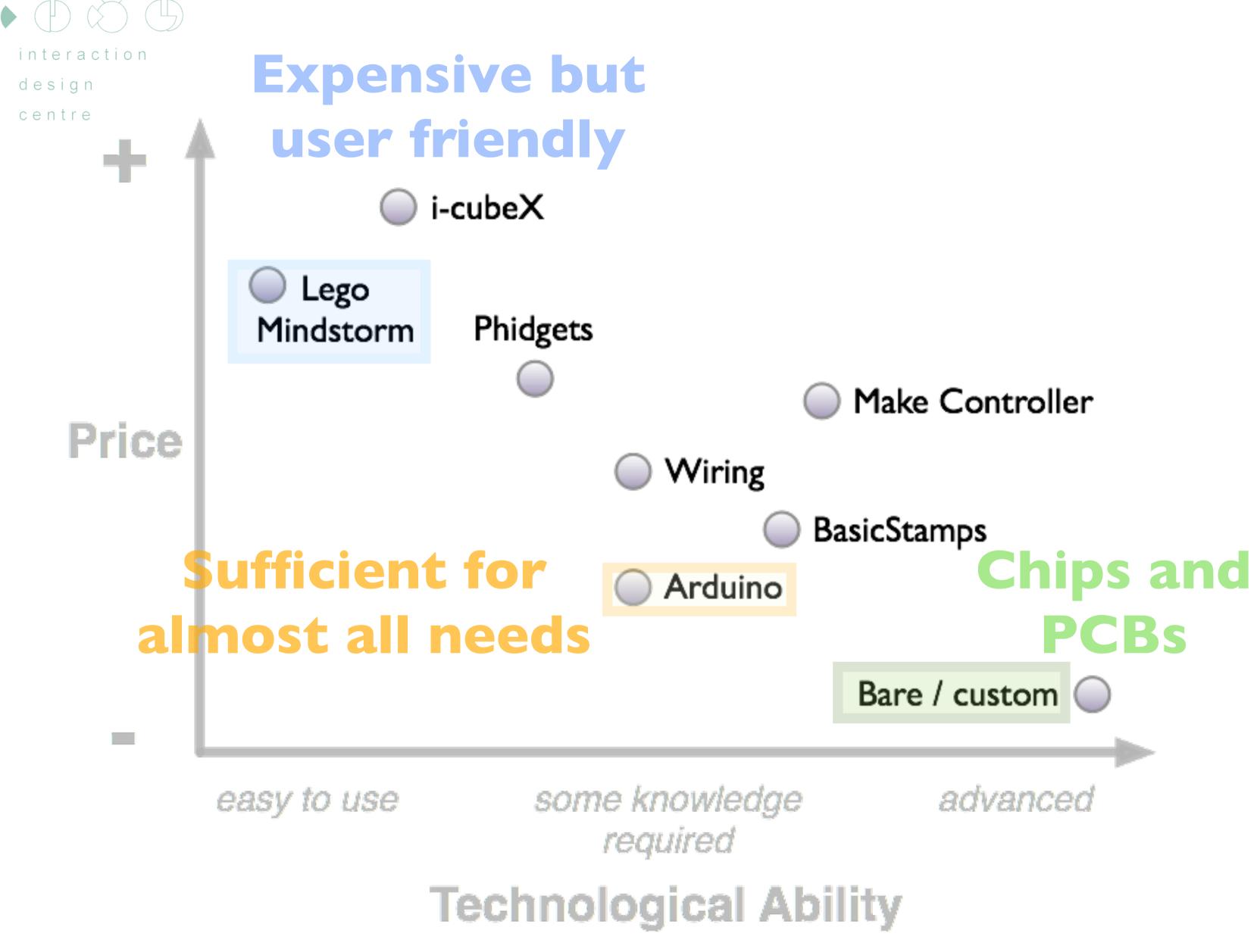






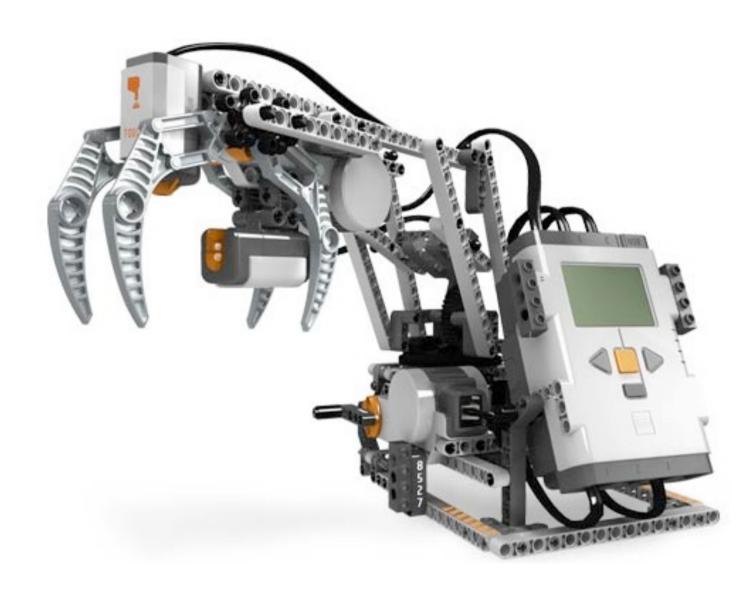




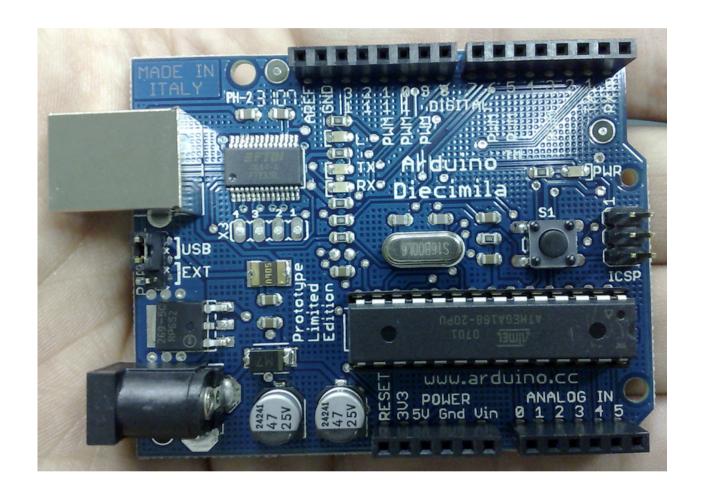








Arduino

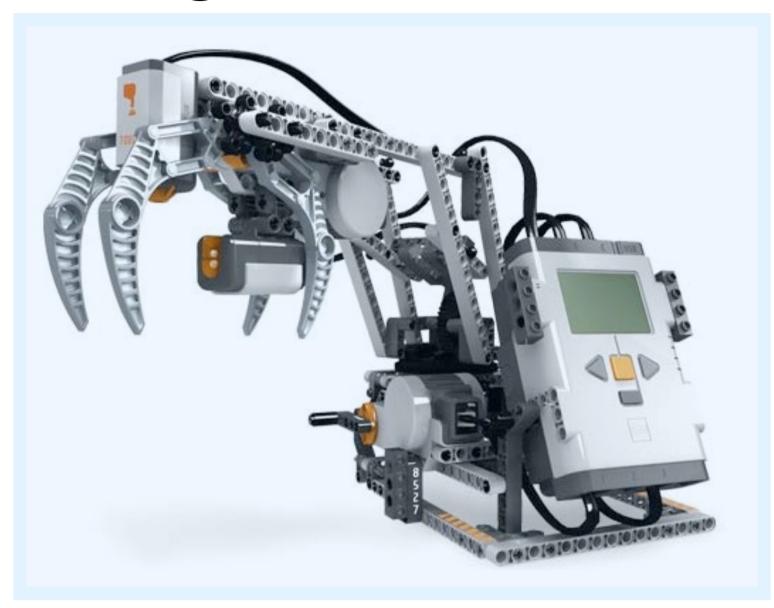


☆ ATMegal68

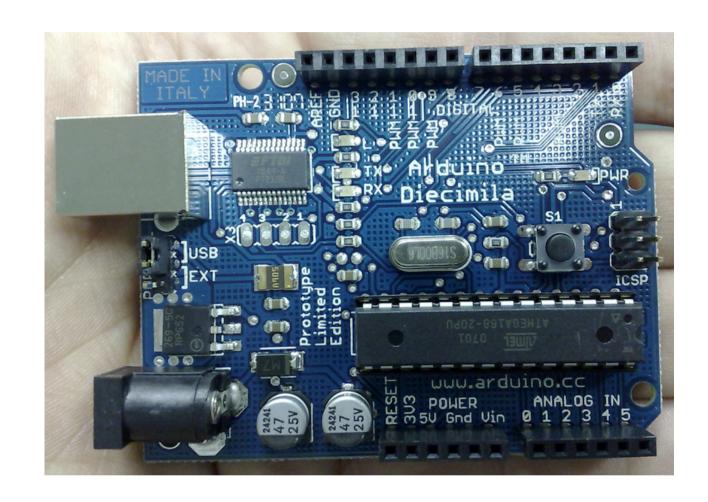












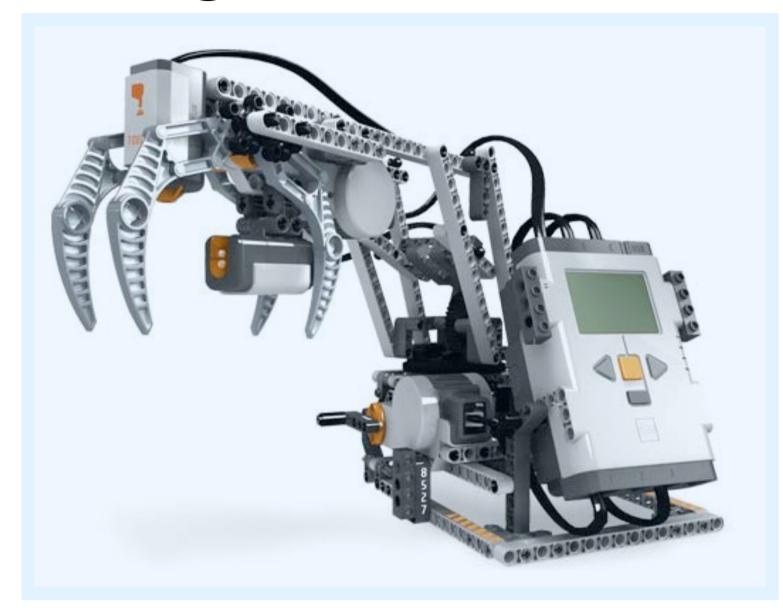
Approx. ~€250

☆ ATMegal68



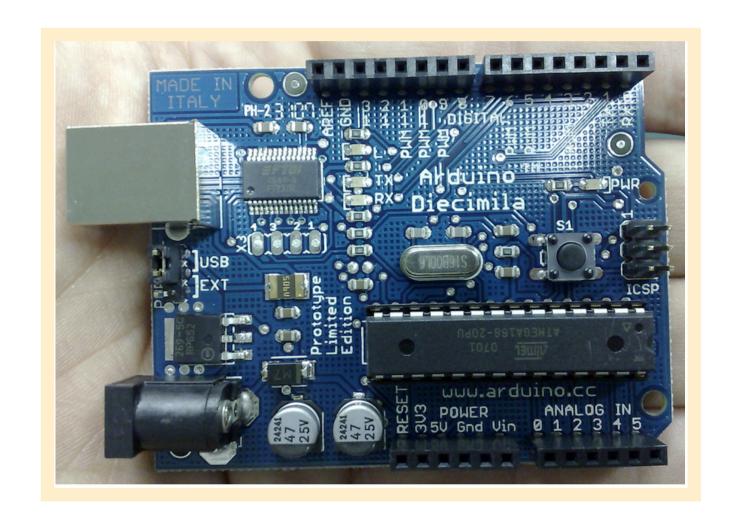












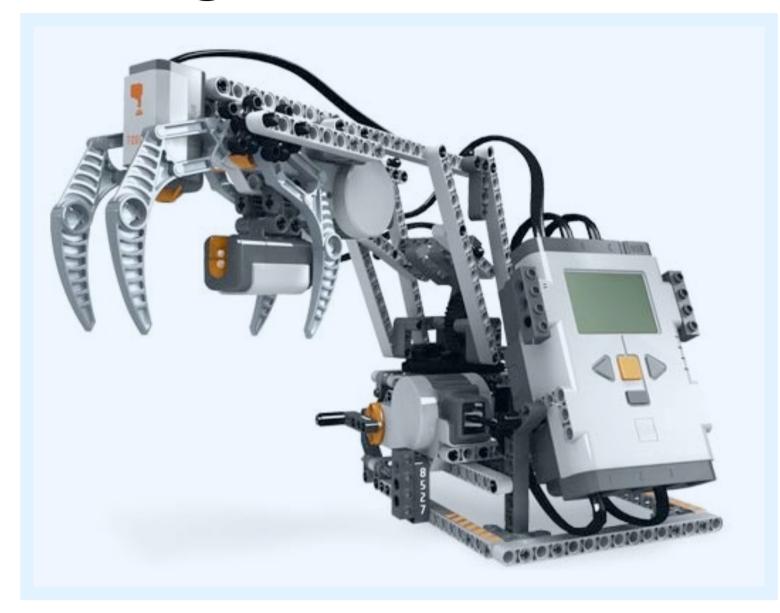
Approx. ~€250

ATMegal 68



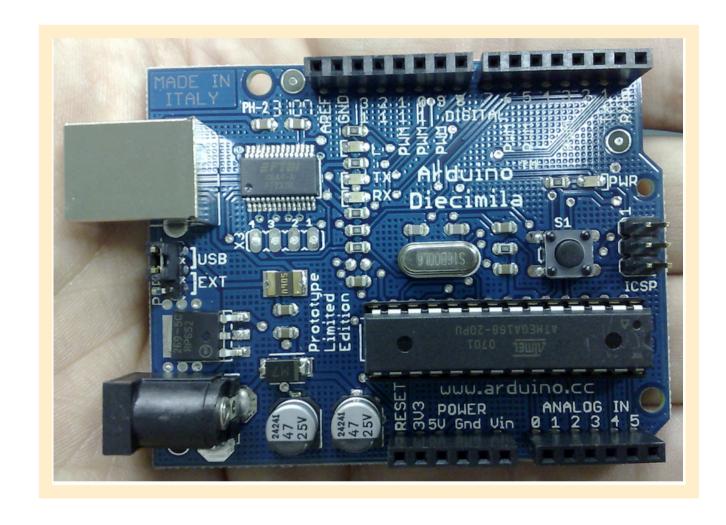






Arduino

Approx. ~€25



Approx. ~€250

ATMegal 68



Approx. ~€4



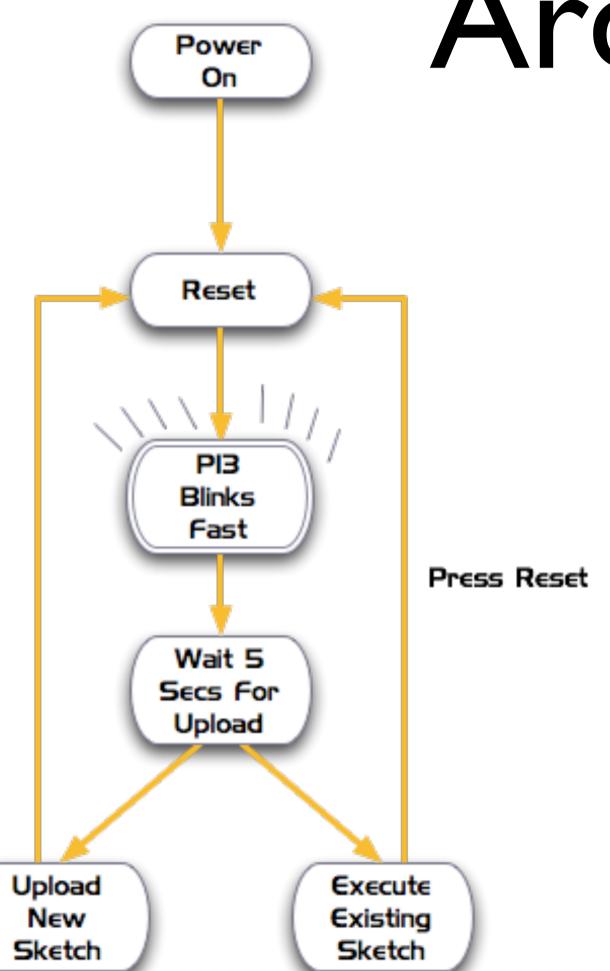


Programming





Programming an Arduino





Compile (check for errors)



Upload to board





An Arduino "Sketch"

Global Variables



Declare variables at top

setup()



Initialize



setup() - run once at beginning, set pins

loop()



Running

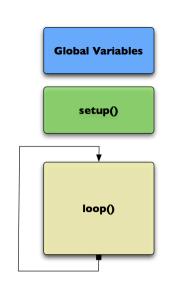






centre

An Arduino "Sketch"



Global Variables

int ledPin = 13; - led connected to control pin 13

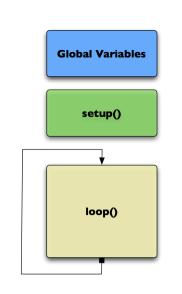
int aSensor = 0; - setup
sensor 'aSensor' on analog pin 0

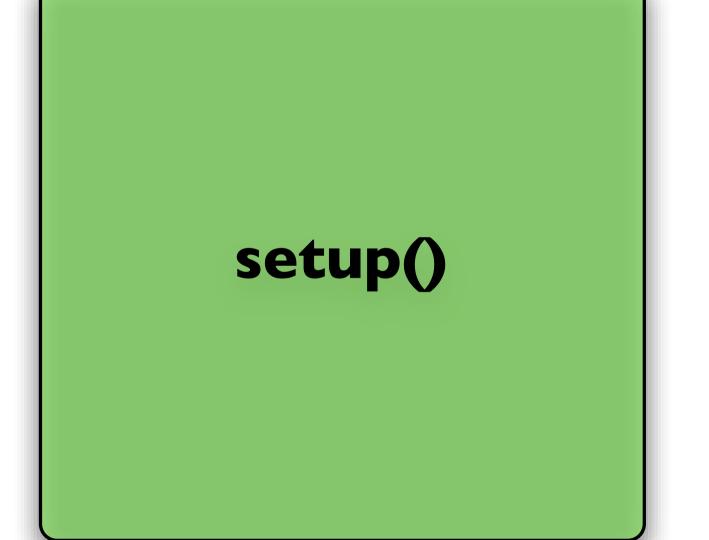
int statePin = LOW; — use this to hold the state of a pin





An Arduino "Sketch"





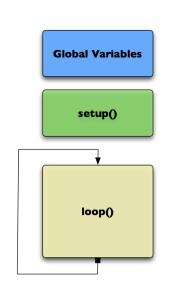
pinlMode() - set a pin as input
or output

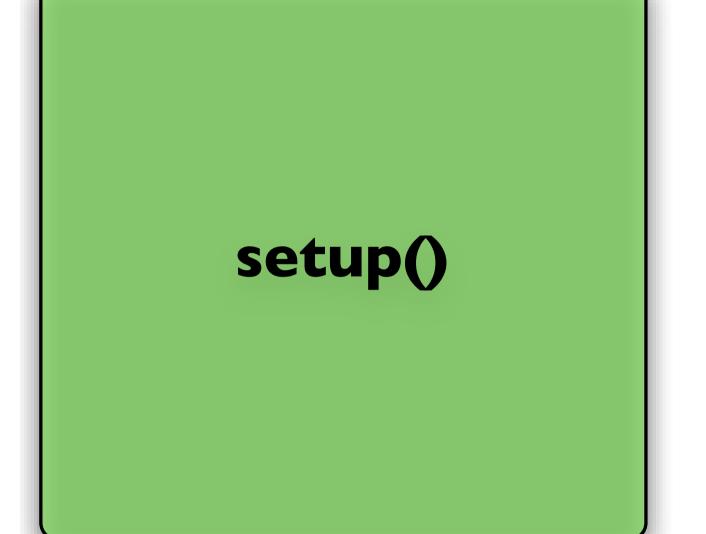
serial.Begin() – setup to `talk' to the computer





An Arduino "Sketch"





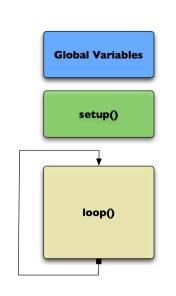
pinlMode(ledPin, Output); – set the pin 'ledPin' as an output

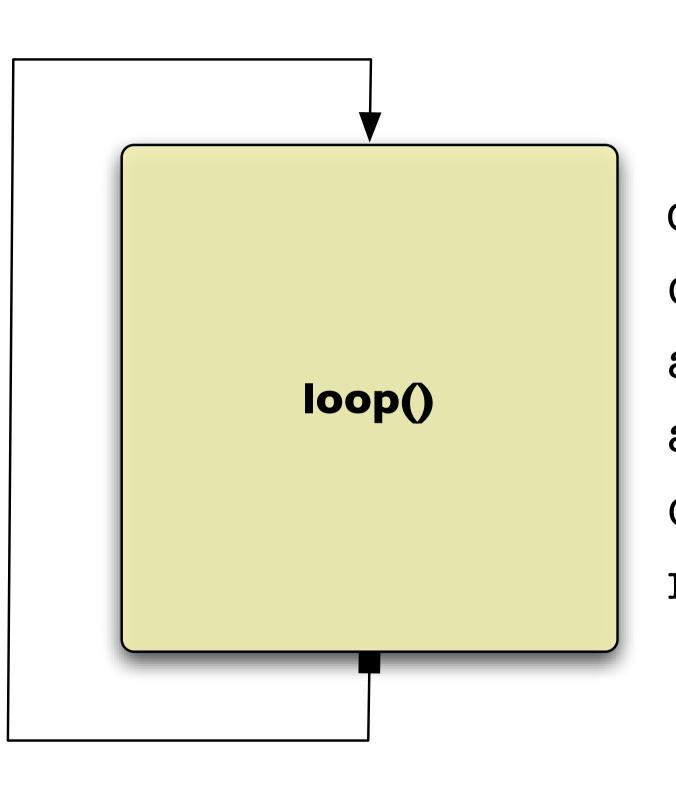
serial.Begin(9600); – talk to the computer at 9600 baud rate





An Arduino "Sketch"



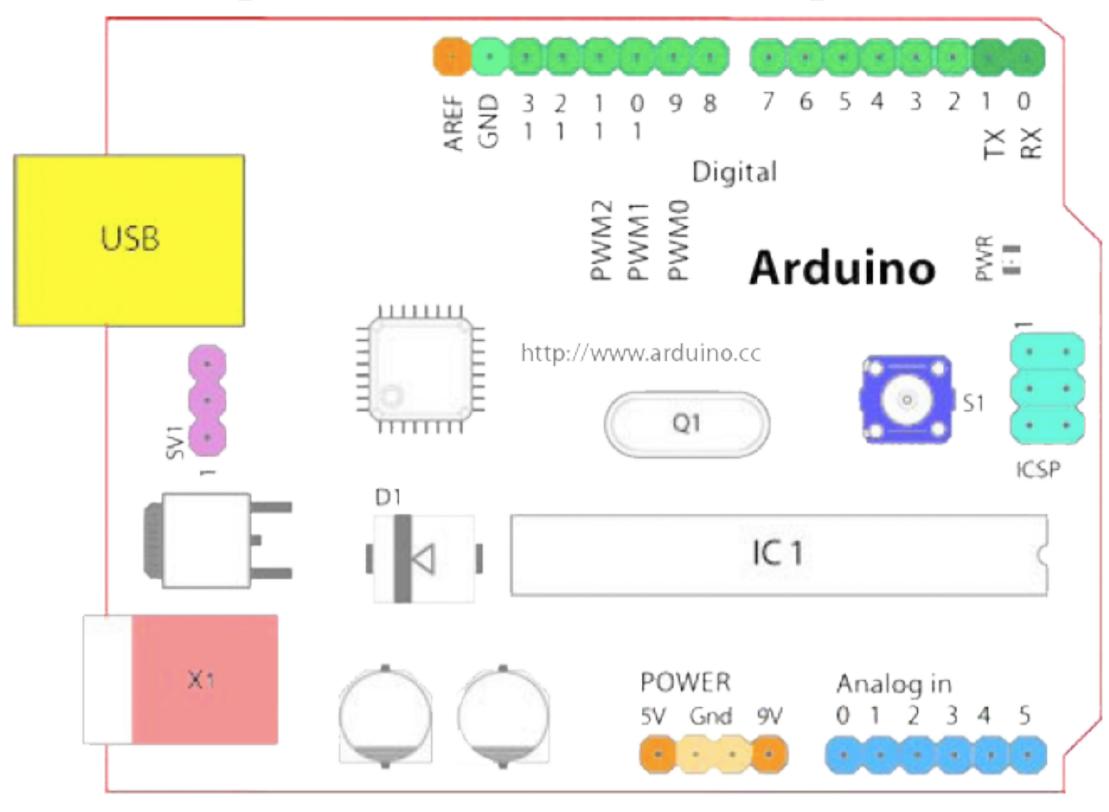


digitalWrite() – set a digital pin high/low digitalRead() – read a digital pin's state analogRead() – read an analog pin analogWrite() – write an "analog" PWM value delay() – wait an amount of time millis() – get the current tim





Input / Output





14 Digital 10 (pins 0 - 13)



6 Analog In (pins 0 - 5)



6 Analog Out (pins 3, 5, 6, 9, 10, 11)





Digital? Analog?

- Digital only has two values: on/off
- Analog has many (infinite) values
- Computers don't really do analog use quantization instead

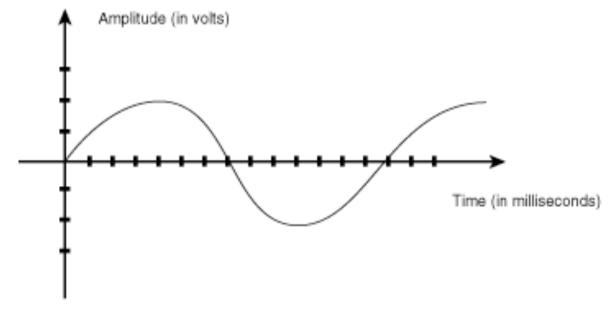




Digital? Analog?

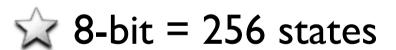
Many states (Analog) or just two (HIGH/LOW - Digital)





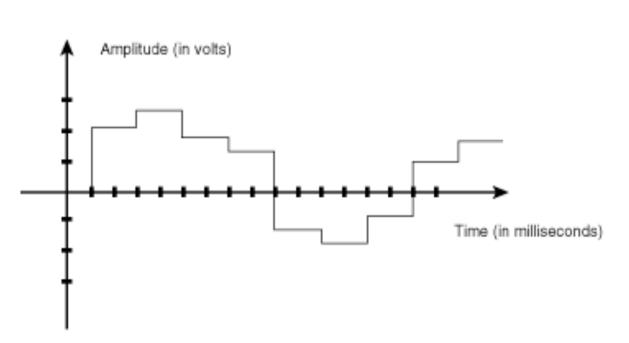


Common computer resolutions:



$$2 \times 16$$
-bit = 65,536 states









10 to/from what?

Sensors



LDR / IR



Switch /

Potentiometer /

Joystick / Piezo



Accelerometer



Viltrasonic

Indicators



ted / Lamps



Buzzers

Actuators



Solenoid



Stepper Motor

Other Circuits



Prototype shields



Special ICs



What is a switch?

What Is a Switch?

Internet of Things Studio, RISD ID 2007



Hello World!

```
void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
}

void loop()
{
  Serial.print("Hello World!\n\r");
  // wait 2sec for next reading:
  delay(2000);
}
```



- 🙀 Install latest Arduino IDE from <u>arduino.cc</u>
- Run Arduino IDE
- Write the code on the left into the editor



- Compile / Verify the code by clicking the play button
- Before uploading your sketch, check the board and the serial port are correct for your Arduino and for your computer

Menu -> Tools -> Board

Menu -> Tools -> Serial Port



Upload the code from the computer to the Arduino using the upload button





centre

```
/* Blinking LED ---
* turns on and off a light emitting diode(LED) connected to a digital
* pin, based on data coming over serial
int ledPin = 13; // LED connected to digital pin 13
int inByte = 0;
void setup()
       pinMode(ledPin, OUTPUT); // sets the digital pin as output
       Serial.begin(19200); // initiate serial communication
void loop()
       while (Serial.available()>0) {
            inByte = Serial.read();
       if (inByte>0) {
            digitalWrite(ledPin, HIGH); // sets the LED on
       } else {
            digitalWrite(ledPin, LOW); // sets the LED off
```





```
/* Blinking LED ---
* turns on and off a light emitting diode(LED) connected to a digital
* pin, based on data coming over serial
                                                             Initialise
int ledPin = 13; // LED connected to digital pin 13
                                                         some of the
int inByte = 0;
                                                            variables
void setup()
       pinMode(ledPin, OUTPUT); // sets the digital pin as output
       Serial.begin(19200); // initiate serial communication
void loop()
       while (Serial.available()>0) {
            inByte = Serial.read();
       if (inByte>0) {
            digitalWrite(ledPin, HIGH); // sets the LED on
       } else {
            digitalWrite(ledPin, LOW); // sets the LED off
```



```
interaction
design
centre
```

```
/* Blinking LED ---
* turns on and off a light emitting diode(LED) connected to a digital
* pin, based on data coming over serial
int ledPin = 13; // LED connected to digital pin 13
int inByte = 0;
void setup()
       pinMode(ledPin, OUTPUT); // sets the digital pin as output
       Serial.begin(19200); // initiate serial communication
void loop()
                                          Setup LED pin and
       while (Serial.available()>0) {
                                           serial connection
           inByte = Serial.read();
       if (inByte>0) {
           digitalWrite(ledPin, HIGH); // sets the LED on
       } else {
            digitalWrite(ledPin, LOW); // sets the LED off
```





```
/* Blinking LED ---
* turns on and off a light emitting diode(LED) connected to a digital
* pin, based on data coming over serial
int ledPin = 13; // LED connected to digital pin 13
int inByte = 0;
void setup()
      pinMode(ledPin, OUTPUT); // sets the digital pin as output
      Serial.begin(19200); // initiate serial communication
                                   oop - Reading the
void loop()
                                serial for info, when
      turn the LEDIon
      if (inByte>0) {
          digitalWrite(ledPin, HIGH); // sets the LED on
      } else {
          digitalWrite(ledPin, LOW); // sets the LED off
```





```
/* Digital reading, turns on and off a light emitting diode (LED) connected to digital
* pin 13, when pressing a pushbutton attached to pin 7. It illustrates the concept of
* Active-Low, which consists in connecting buttons using a 1K to 10K pull-up resistor.
int ledPin = 13; // choose the pin for the LED
int inPin = 7; // choose the input pin (button)
int buttonval = 0; // variable for reading the pin status
void setup() {
       pinMode(ledPin, OUTPUT); // set LED as output
       pinMode(inPin, INPUT); // set pushbutton as input
       Serial.begin(19200); // start serial communication to computer
void loop() {
       buttonval = digitalRead(inPin); // read the pin and get the button's state
       if (buttonval == HIGH) { // check if the input is HIGH (button released)
            digitalWrite(ledPin, LOW); // turn LED OFF
            Serial.write('0'); // Button off (0) sent to computer
       } else {
            digitalWrite(ledPin, HIGH); // turn LED ON
            Serial.write('1'); // Button on (1) sent to computer
```



```
interaction
design
centre
```

```
/* Digital reading, turns on and off a light emitting diode (LED) connected to digital
* pin 13, when pressing a pushbutton attached to pin 7. It illustrates the concept of
* Active-Low, which consists in connecting buttons using a 1K to 10K pull-up resistor.
                                                                 Initialise
int ledPin = 13; // choose the pin for the LED
                                                             some of the
int inPin = 7; // choose the input pin (button)
int buttonval = 0; // variable for reading the pin status
                                                                variables
void setup() {
       pinMode(ledPin, OUTPUT); // set LED as output
       pinMode(inPin, INPUT); // set pushbutton as input
       Serial.begin(19200); // start serial communication to computer
void loop() {
       buttonval = digitalRead(inPin); // read the pin and get the button's state
       if (buttonval == HIGH) { // check if the input is HIGH (button released)
            digitalWrite(ledPin, LOW); // turn LED OFF
            Serial.write('0'); // Button off (0) sent to computer
       } else {
            digitalWrite(ledPin, HIGH); // turn LED ON
            Serial.write('1'); // Button on (1) sent to computer
```



```
interaction
design
centre
```

```
/* Digital reading, turns on and off a light emitting diode (LED) connected to digital
* pin 13, when pressing a pushbutton attached to pin 7. It illustrates the concept of
* Active-Low, which consists in connecting buttons using a 1K to 10K pull-up resistor.
                                                           Setup LED pin,
int ledPin = 13; // choose the pin for the LED
int inPin = 7; // choose the input pin (button)
                                                           switch pin and
int buttonval = 0; // variable for reading the pin status
void setup() {
       pinMode(ledPin, OUTPUT); // set LED as output
       pinMode(inPin, INPUT); // set pushbutton as input
       Serial.begin(19200); // start serial communication to computer
void loop() {
       buttonval = digitalRead(inPin); // read the pin and get the button's state
       if (buttonval == HIGH) { // check if the input is HIGH (button released)
            digitalWrite(ledPin, LOW); // turn LED OFF
            Serial.write('0'); // Button off (0) sent to computer
       } else {
            digitalWrite(ledPin, HIGH); // turn LED ON
            Serial.write('1'); // Button on (1) sent to computer
```



```
interaction
design
centre
```

```
/* Digital reading, turns on and off a light emitting diode (LED) connected to digital
* pin 13, when pressing a pushbutton attached to pin 7. It illustrates the concept of
* Active-Low, which consists in connecting buttons using a 1K to 10K pull-up resistor.
                                                 Loop - Reading the
int ledPin = 13; // choose the pin for the LED
                                               button for info, when
int inPin = 7; // choose the input pin (button)
int buttonval = 0; // variable for reading the pin status
                                                button is press turn
void setup() {
       pinMode(ledPin, OUTPUT); // set LED as output
       pinMode(inPin, INPUT); // set pushbutton as input
       Serial.begin(19200); // start serial communication to computer
void loop() {
       buttonval = digitalRead(inPin); // read the pin and get the button's state
       if (buttonval == HIGH) { // check if the input is HIGH (button released)
            digitalWrite(ledPin, LOW); // turn LED OFF
            Serial.write('0'); // Button off (0) sent to computer
       } else {
            digitalWrite(ledPin, HIGH); // turn LED ON
            Serial.write('1'); // Button on (1) sent to computer
```





d e s i g n c e n t r e

Spooky Skull Example







centre

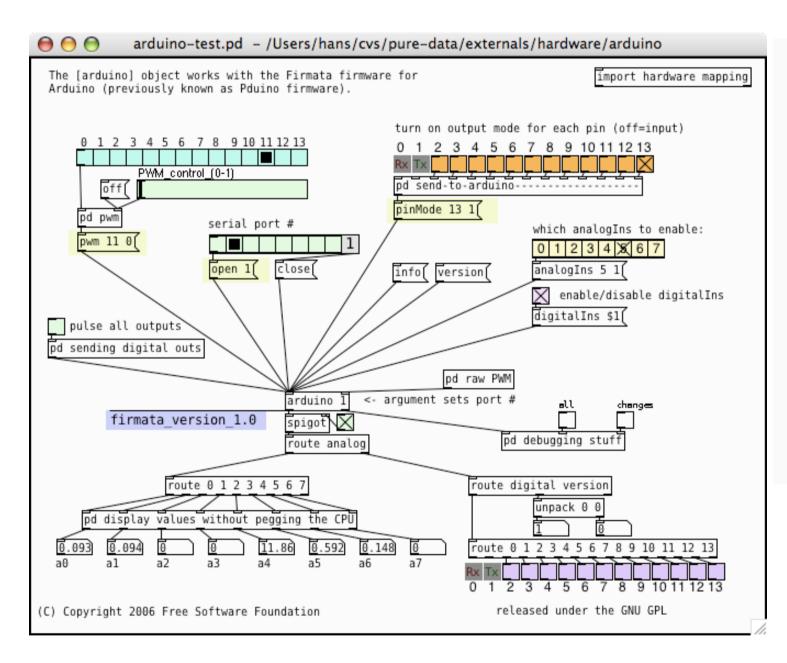
To/Erom Other Software





interaction design centre

Pure Data



```
start polling
                                                               have fun!
metro 40
114, 32, 100, 13 the "poll" command
  open 3
  close(
comport 3 115200 port number 3 (on my macbook, Yours might be a different
pd make my_list
unpack f f f f f f f f f f f f f f f f
                                    \boxtimes digital in 2 - 12
            478 analog in 5
         510 analog in 4
       515 analog in 3
     519 analog in 2
  498 analog in 1
501 analog in 0
```









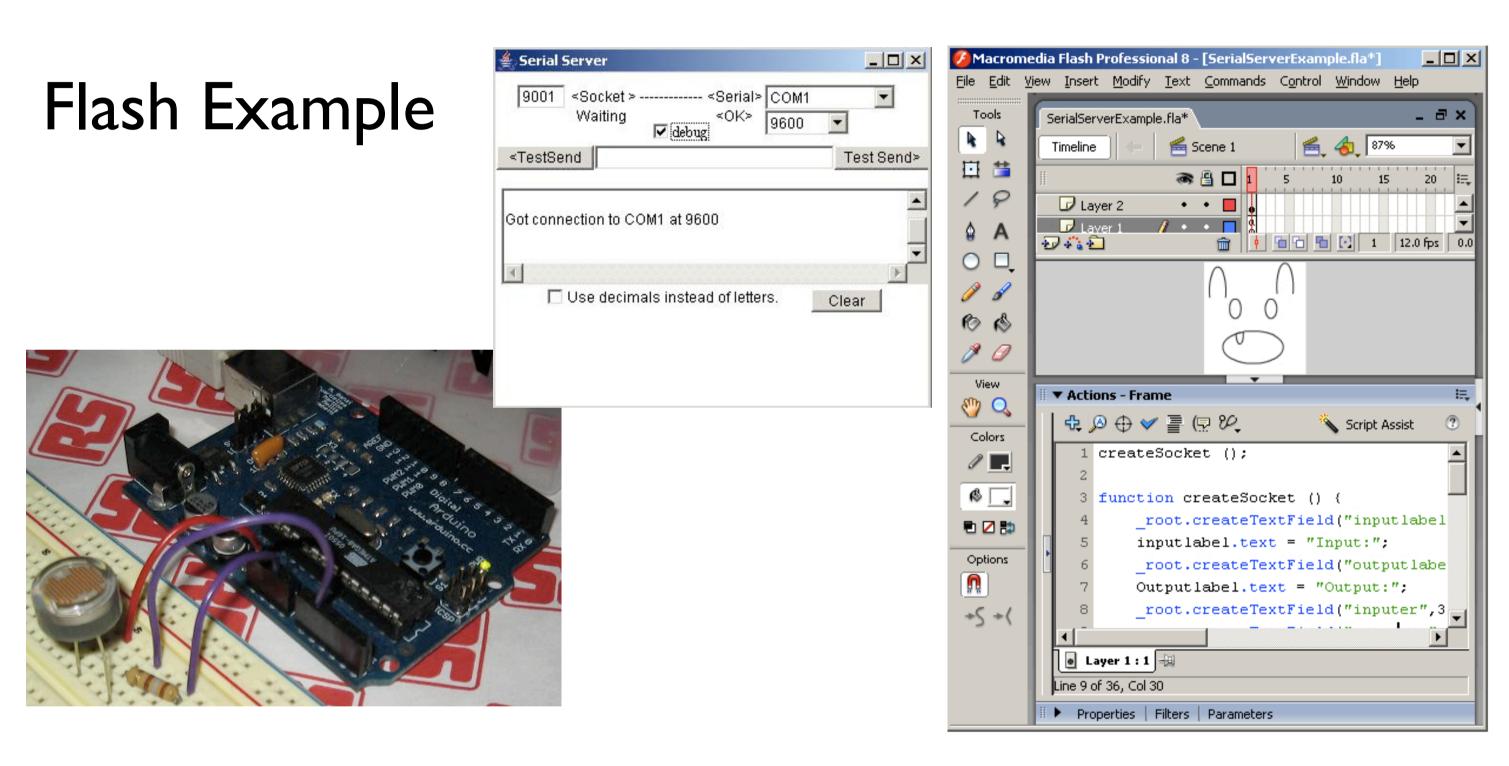
Flash

★ Serial Proxy
(XMLSocket server) a
small program that runs
on your Mac/PC and
keeps a live connection
between the serial port
and Flash

```
// arduino serial proxy helper
// send messages from arduino to your
// serial port >> socket
// and read it through flash
#define inPin 2
void setup(){
Serial.begin(9600);
void loop(){
 byte readValue=digitalRead(inPin);
 if(readValue == HIGH){
 Serial.print("high");
 Serial.print(0, BYTE);
```







http://file-error.net/IoIoIoIoI/?Physical_Computing_and_Interaction:Arduino:Arduino_VS_Flash_http://protolab.pbwiki.com/Arduino2Flash_





interaction design centre

Animata

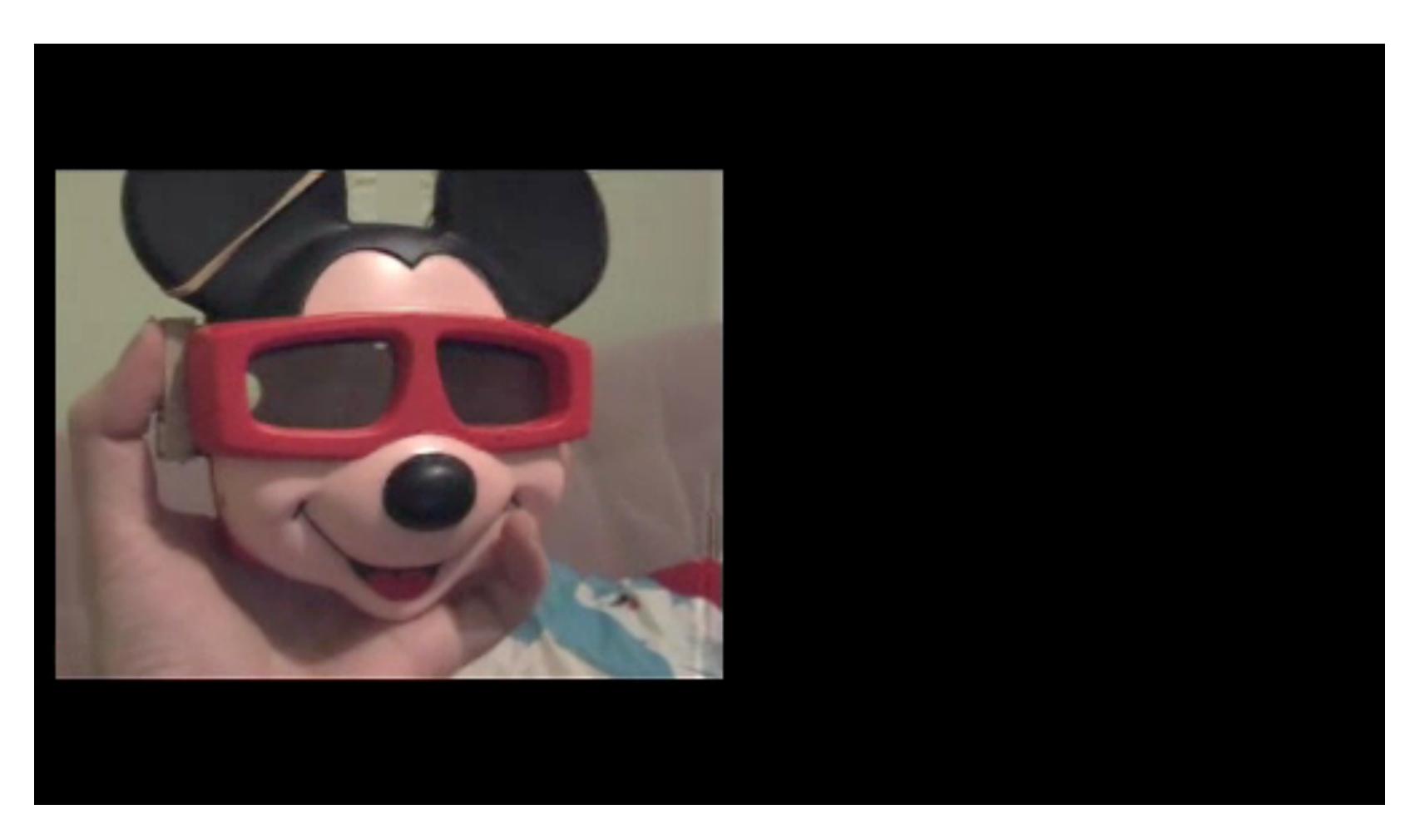






interaction design centre

Animata







centre

Processing

```
000
                                           BouncingBall | Processing 1.0.1
File Edit Sketch Tools Help
  BouncingBall §
                  Ball
Ball ball;
void setup() {
 size(320,240);
 ball= new Ball(0,0);
 background(0);
noStroke();
 smooth();
void draw() {
 background(0);
 ball.calc();
 ball.draw();
```





design design Interactive Objects / Displays



- Spooky Skull Example Tod E. Kurt
- Lilypad Leah Buechley
- Wanderlust Hector Ouilhet
- Ambient Auditory Group Presence Display
- A Hanging Gardens

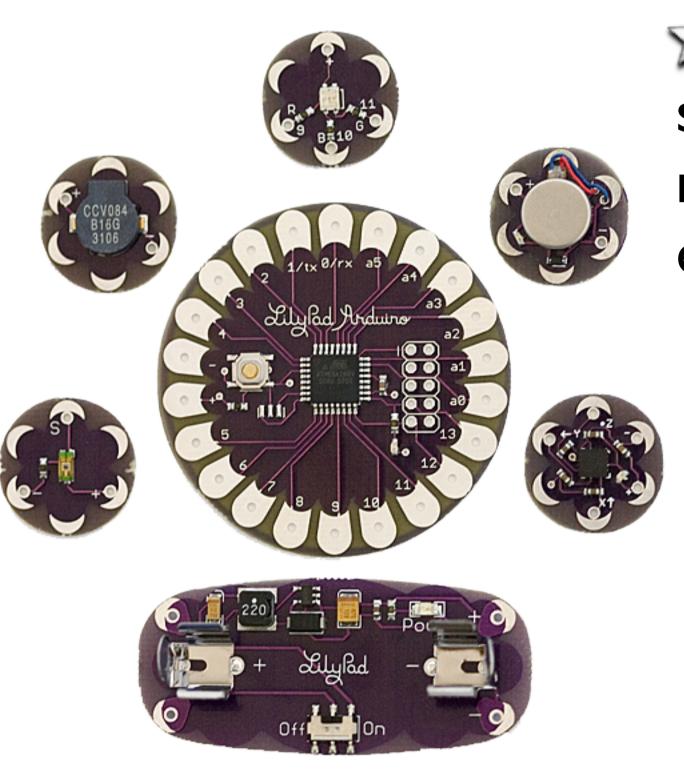


💢 Dawn07, Dawn08, Dawn 09 (What will you do ?)





Lilypad



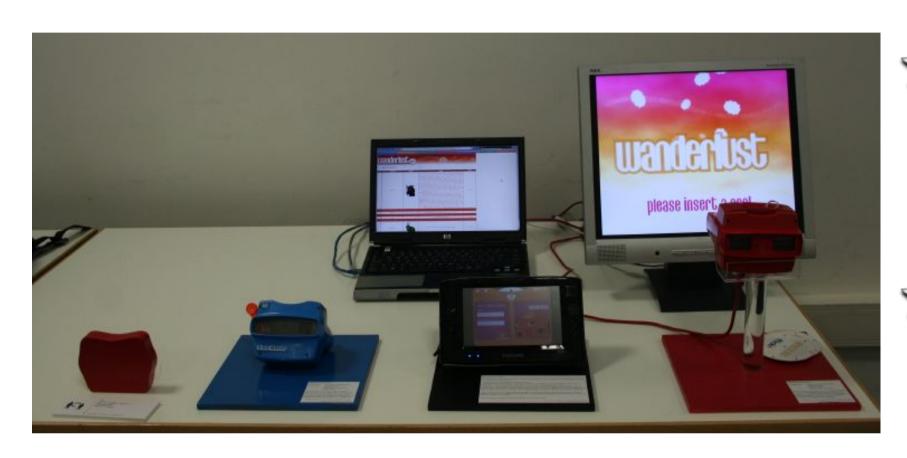
A set of stitchable controllers, sensors and actuators enables novices to build their own electronic textiles.







Wanderlust

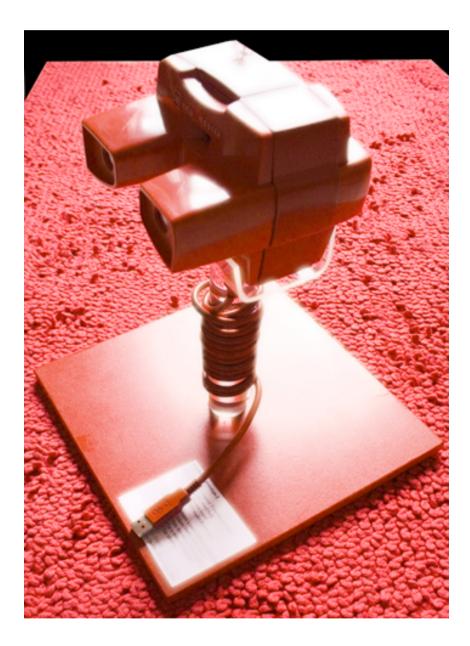


- A mobile service, uses identities and needs (in terms of time and distance) to explore spaces as well to meet people with similar interests.
- Ad-Hoc Social Networking tool as it provides suggestions from people and places that can be relevant to you.
- The prototype used physical tokens (reels) to represent a user's different identities. Each reel is mapped to an identity; when this reel is introduced to the View-Master it will play a video showing the contents and attributes of this identity.
- Consists of a original View-Master (red), wooden body, Arduino board, RFID reader & antenna

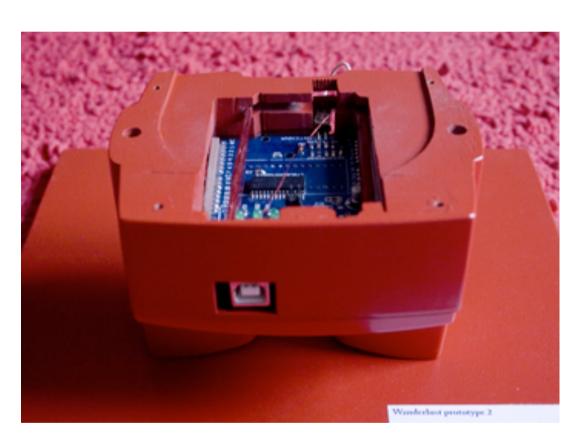




Wanderlust Prototype





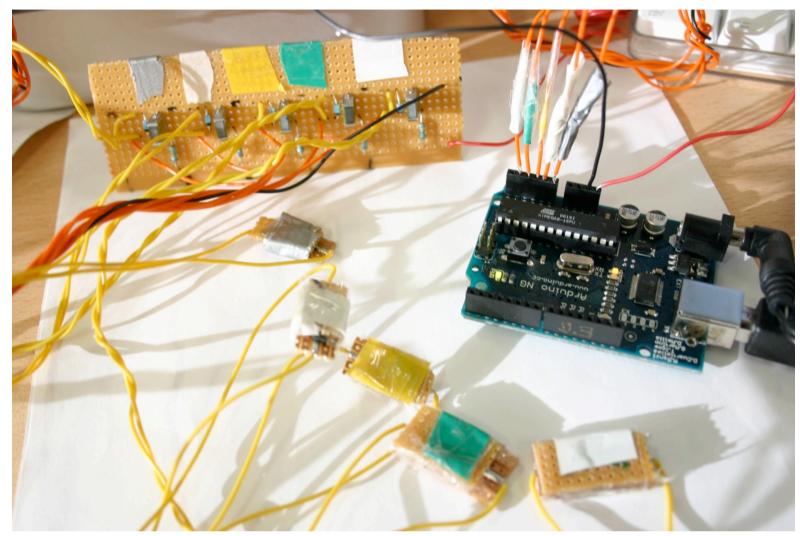


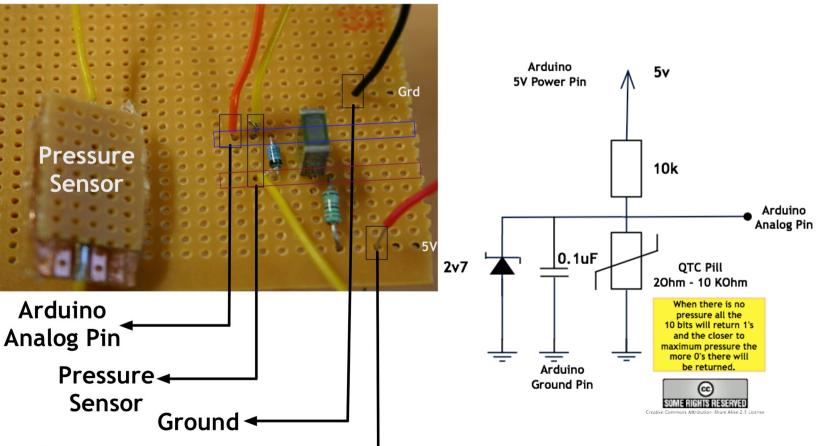




interaction
design
centre

Arduino 5V Power Pin





Ambient Auditory Group Presence Display:

- Designed to explore, using concurrent auditory icons, the issue of group awareness (ICAD07 paper)
- Presence and activity are conveyed by changes in the soundscape
- Ruby, Python, C, Growl and Boodler
- Semi-Public Display for Small, Colocated Groups





interaction

a e

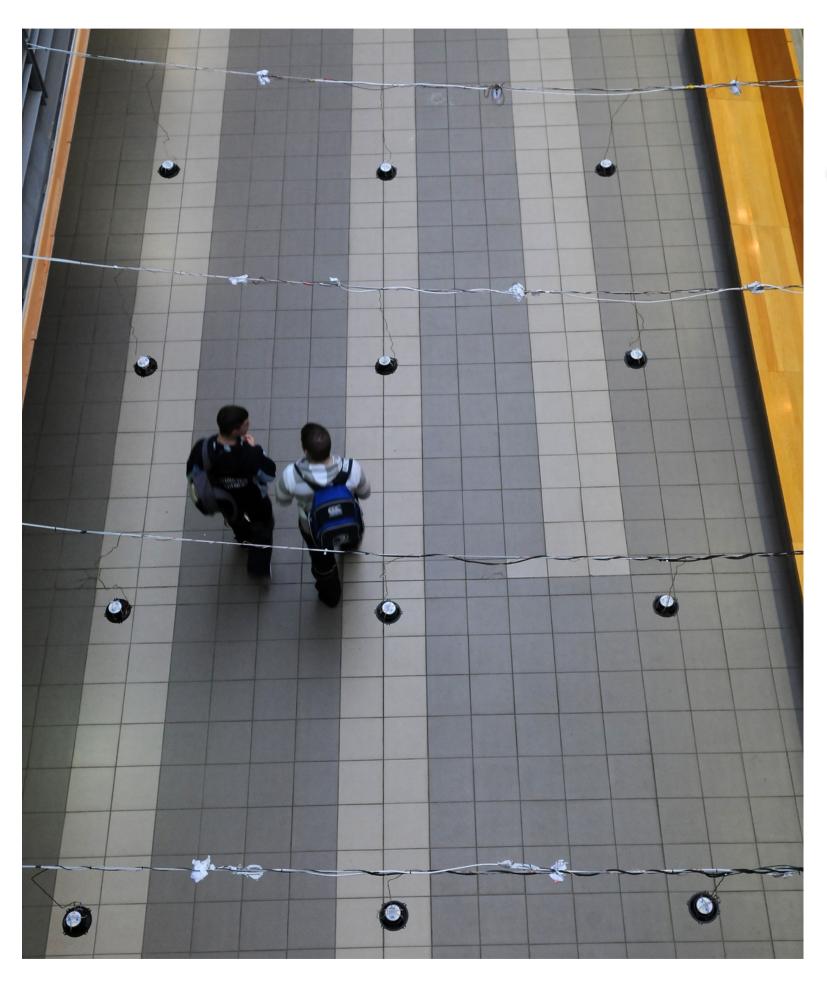
Hanging Gardens Two Places Exhibition

Eoin Brazil and Jürgen Simpson





design centre





- Collaboration with Jurgen Simpson
- Two Places UL / Ormeau, Belfast
- Network of Speakers and Sensors
- Arduino, Ruby, Max/MSP
- ☆ 2 field of insects
- Circadian rhythm
- Walls and nodes





Useful Links

- Arduino http://www.arduino.cc/
- Wanderlust http://www.thejaguarhouse.com/blog/
- Arduino lectures http://www.slideshare.net/eoinbrazil
- Tod E. Kurt's blog (check his Spooky Arduino projects) http://todbot.com/
 blog/category/arduino/
- x ITP Physical Computing http://itp.nyu.edu/physcomp/Intro/HomePage
- The Art and Craft of Toy Design http://yg.typepad.com/makingtoys2/
- 🙀 Lilypad http://www.cs.colorado.edu/~buechley/diy/diy_lilypad_arduino.html
- www. Usman Haque and Adam Somlai-Fischer ``Low tech sensors and actuators for artists and architects"

