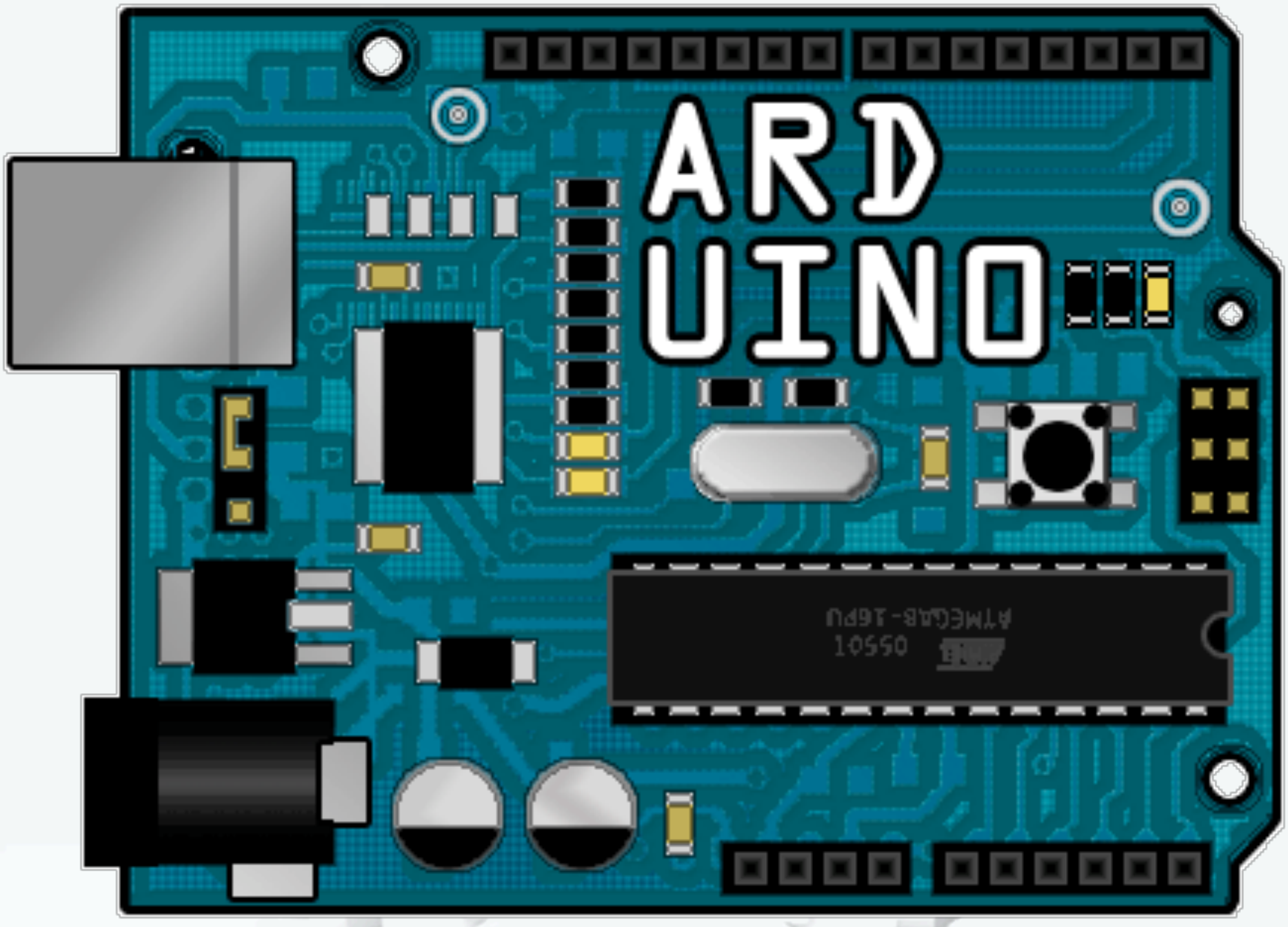


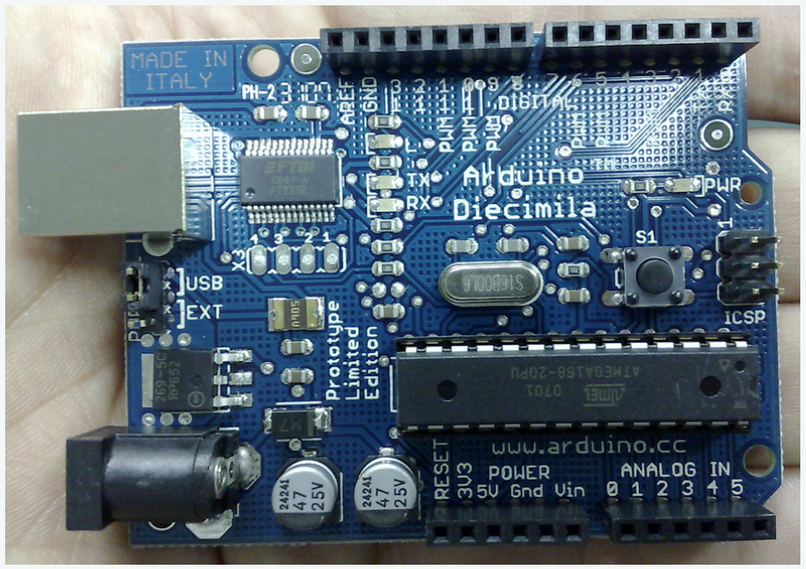

http://www.flickr.com/photos/collinmel/2317520331/

# What is Arduino?

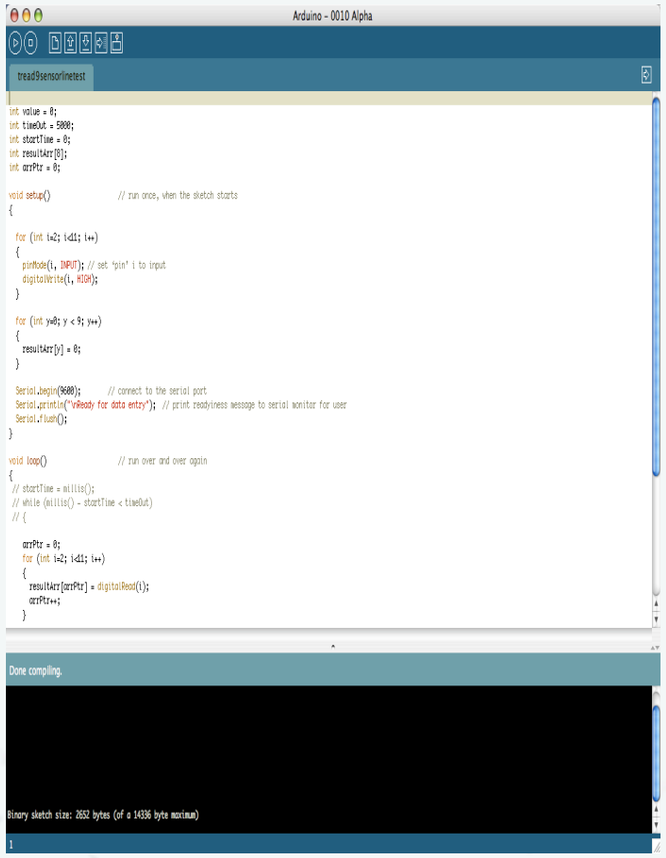
interaction design centre

irish open source technology conference

## The hardware




## The development environment




## The community

# Why Arduino?

interaction design centre

- *artists & designers*
- *"opportunistic prototyping"*
  - device hacking & reuse
- *"open source hardware"*
  - Open Source Physical Computing Platform
- open source
  - free to inspect & modify
- community
  - wiki, forums, tutorials

# The Nuts & Bolts

interaction design centre

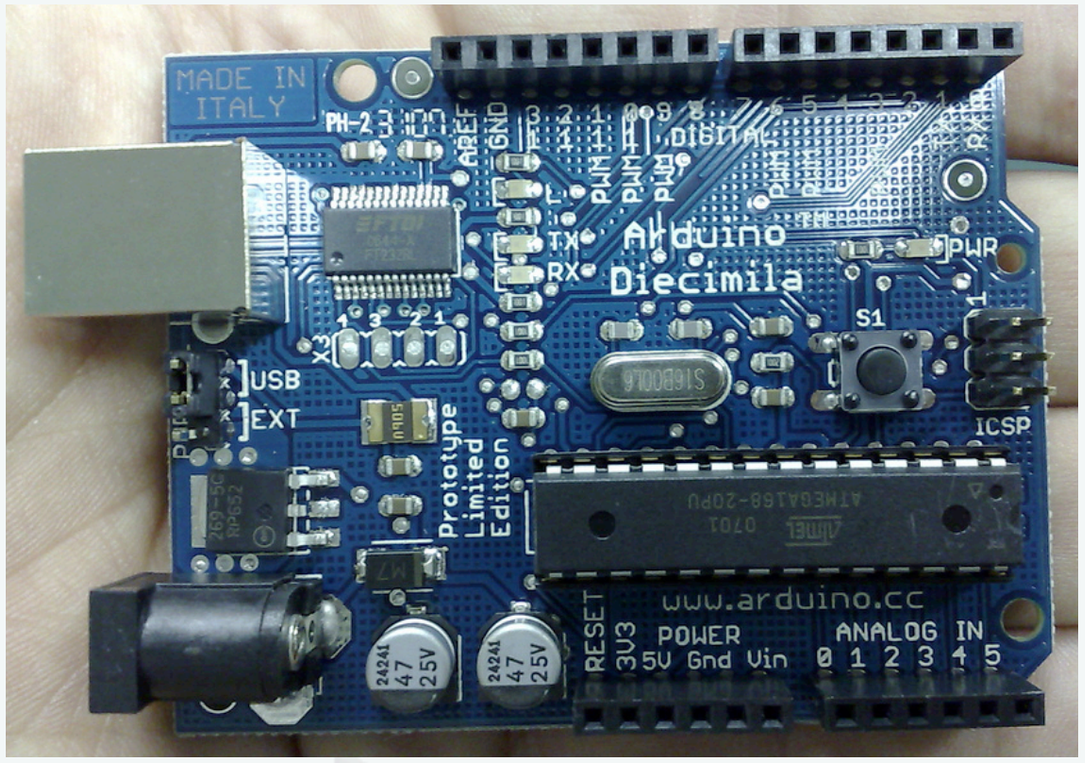
- physical computing. er, what? ubiquitous computing, pervasive computing, ambient intelligence, calm computing, everyware, spimes, blogjects, smart objects...
- tiny computer you can program
- completely stand-alone, talks to other devices

- 'C'
- Flash
- Processing
- PD
- Max/MSP
- Ruby
- Python
- PHP
- Matlab
- Squeak (Smalltalk)

# Arduino Capabilities

interaction design centre

irish open source technology conference

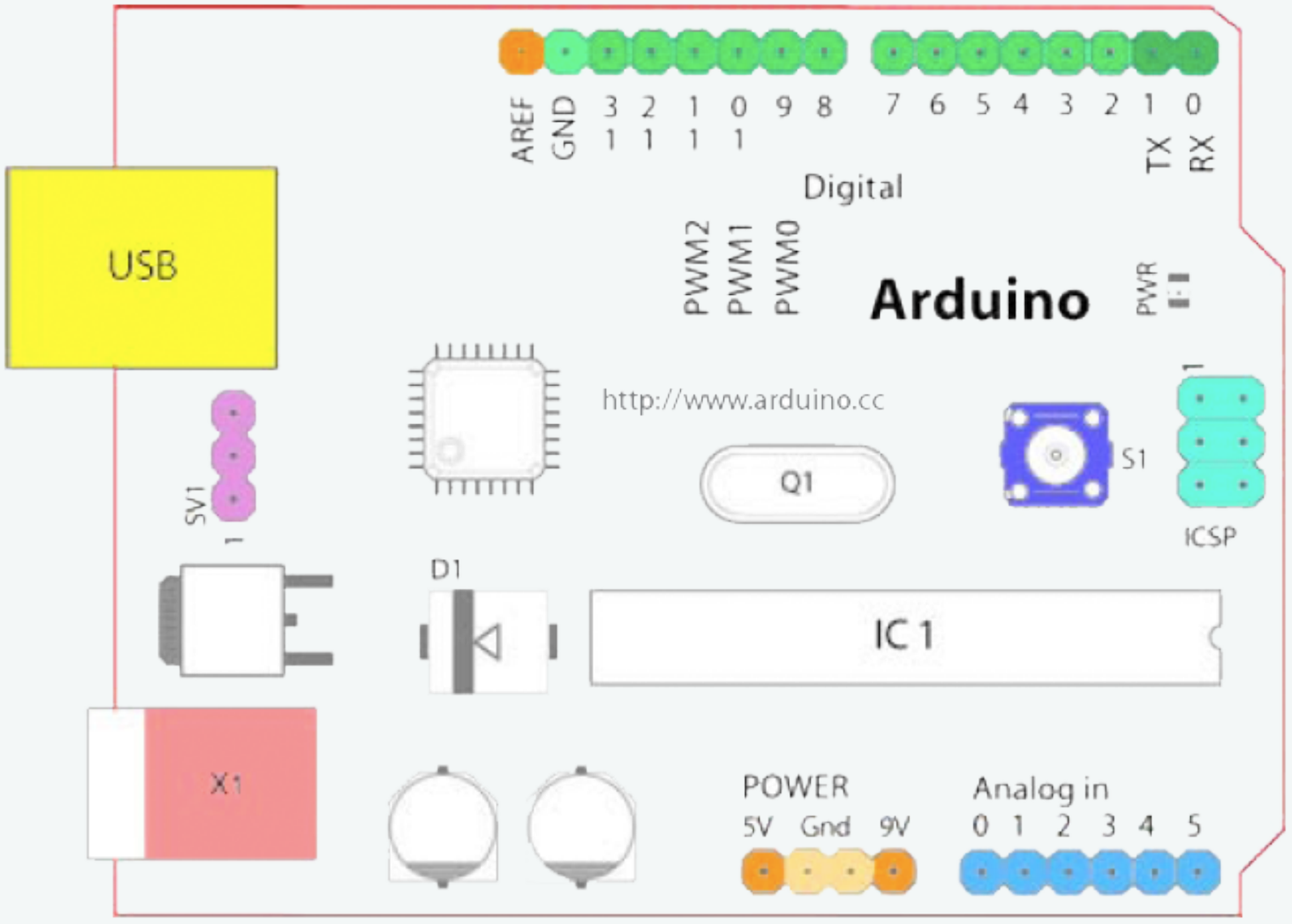



Intel 286

Arduino

# Layout of an Arduino






- ☆ 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 communiation (e.g. Serial.begin).

- ☆ Reset Button - S1 (dark blue)
- ☆ In-circuit Serial Programmer (blue-green)
- ☆ Analog Reference pin (orange)
- ☆ Analog In Pins 0-5 (light blue)
- ☆ Power and Ground Pins (power: orange, grounds: light orange)
- ☆ External Power Supply In (9-12VDC) - X1 (pink)

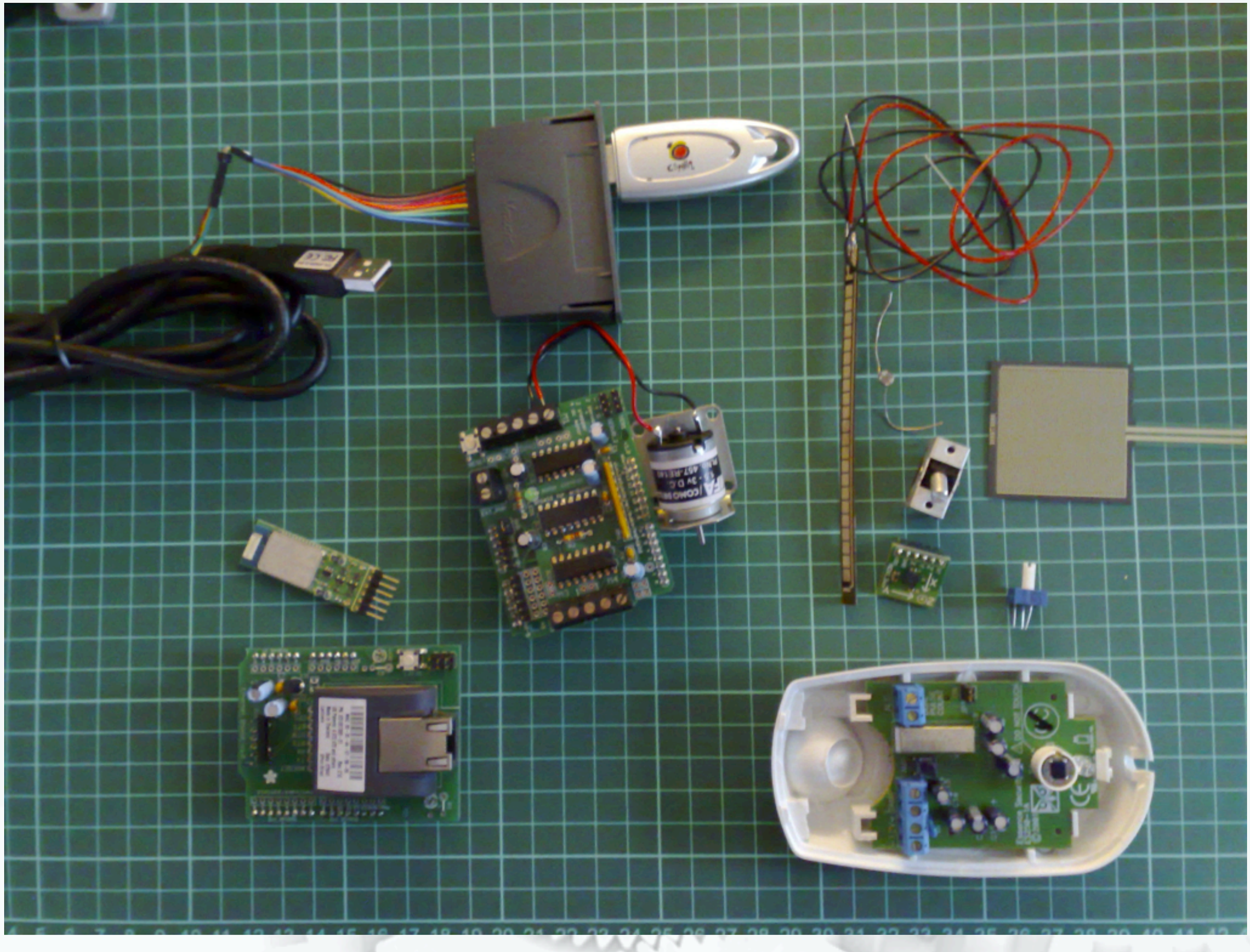
- ☆ Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- ☆ USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

# 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*'' - 1 (HIGH) or 0 (LOW) value (i.e. on/off)
- ``*analog*'' - range (0-255 typically), e.g. LED brightness

# Arduino Connections





Bluetooth - BlueSmirf

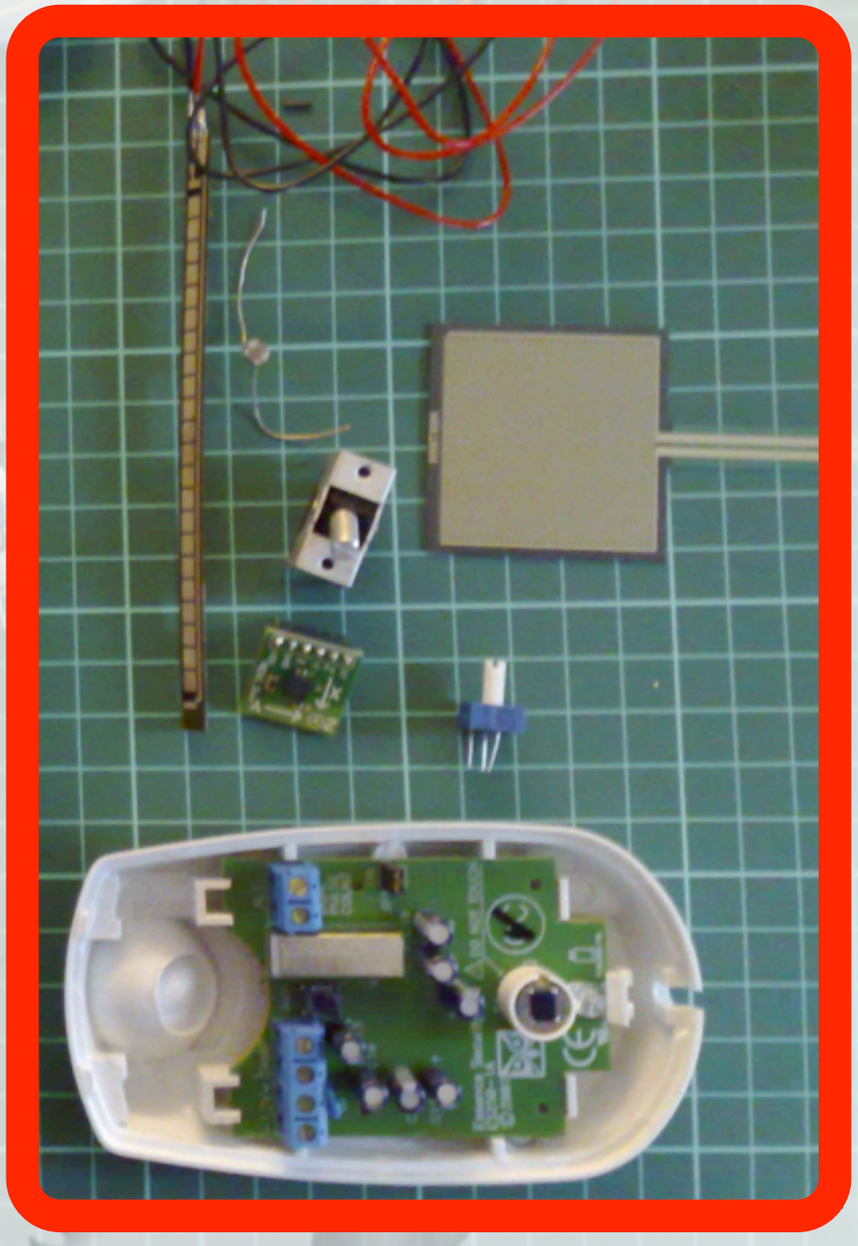
Internet - MatchPort

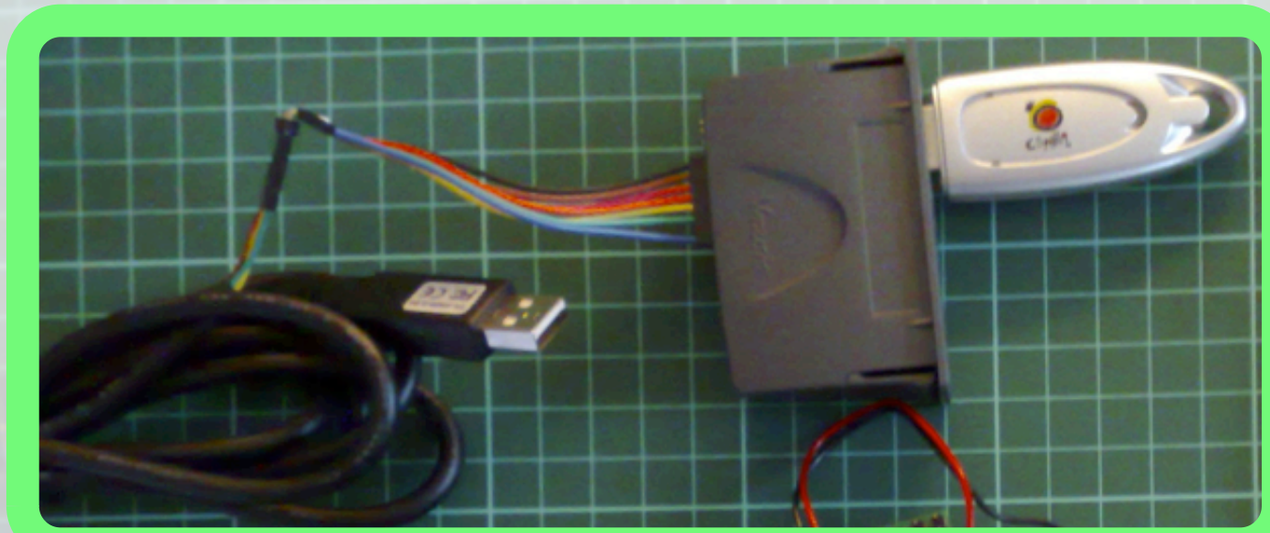
Many others:

Wifi, IrDa, Zigbee, etc.

# Motors:
# DC, Steppers, Servos

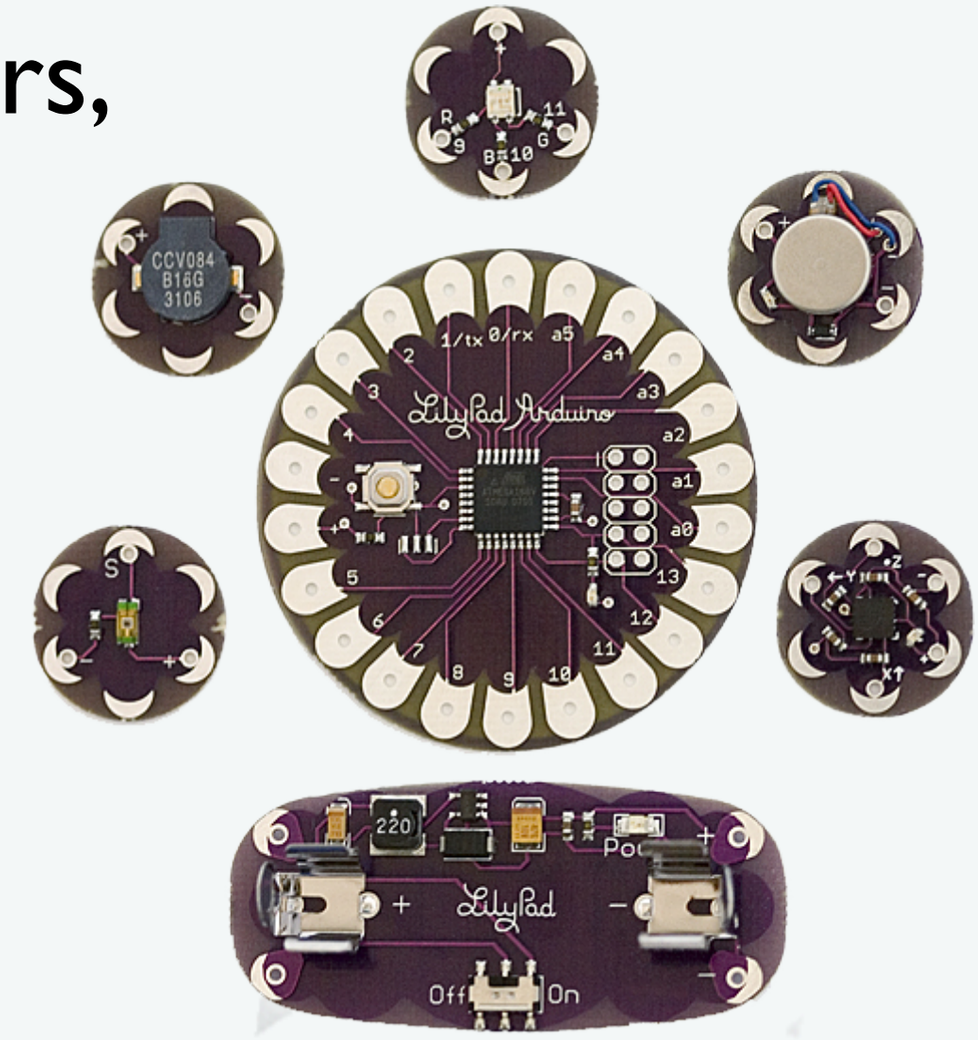
# Sensors:
# Flex, IrDa, Switches, FSR, Accelerometers





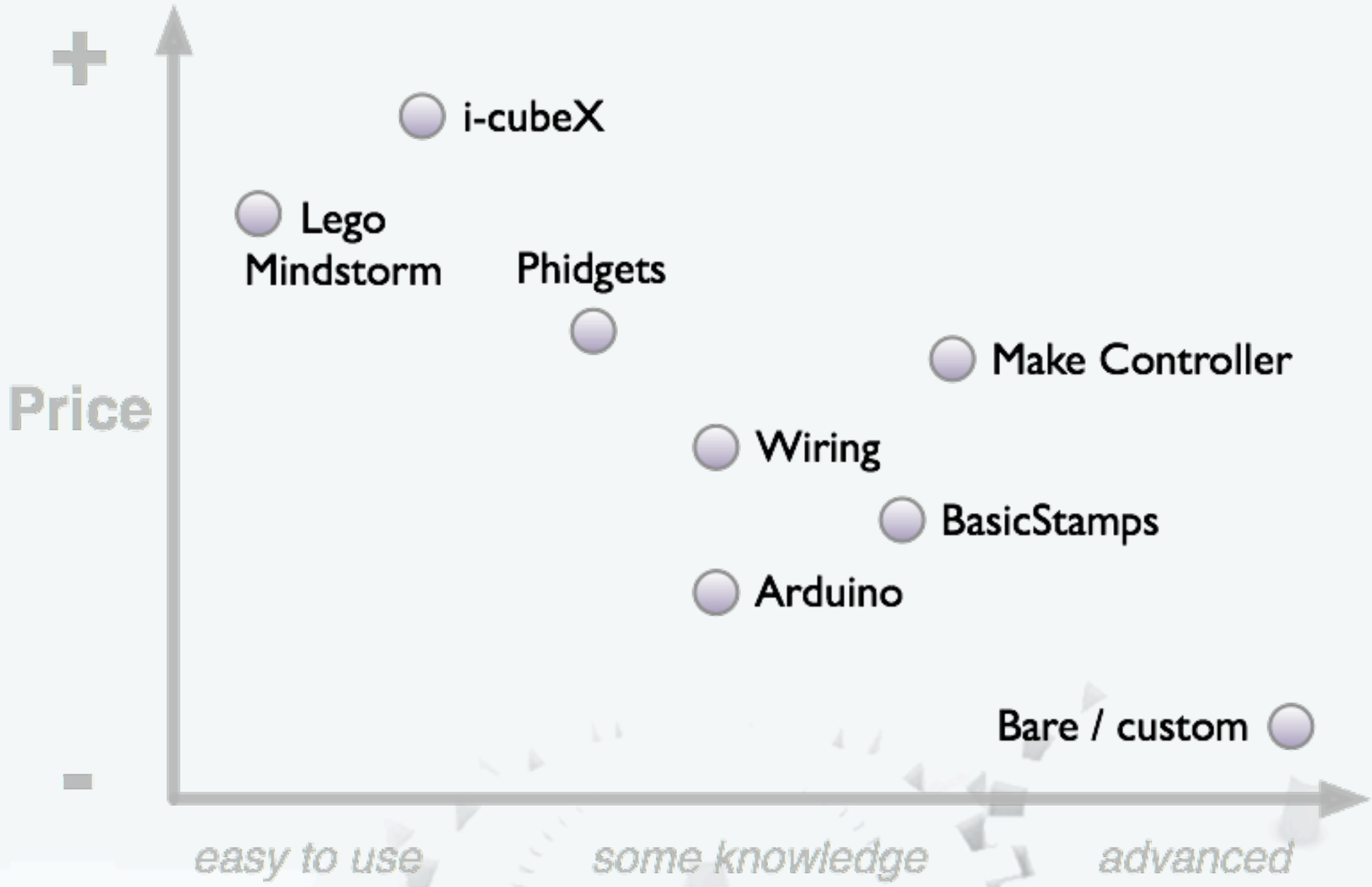
# Custom Hardware: e.g. VMusic 2 MP3 player

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



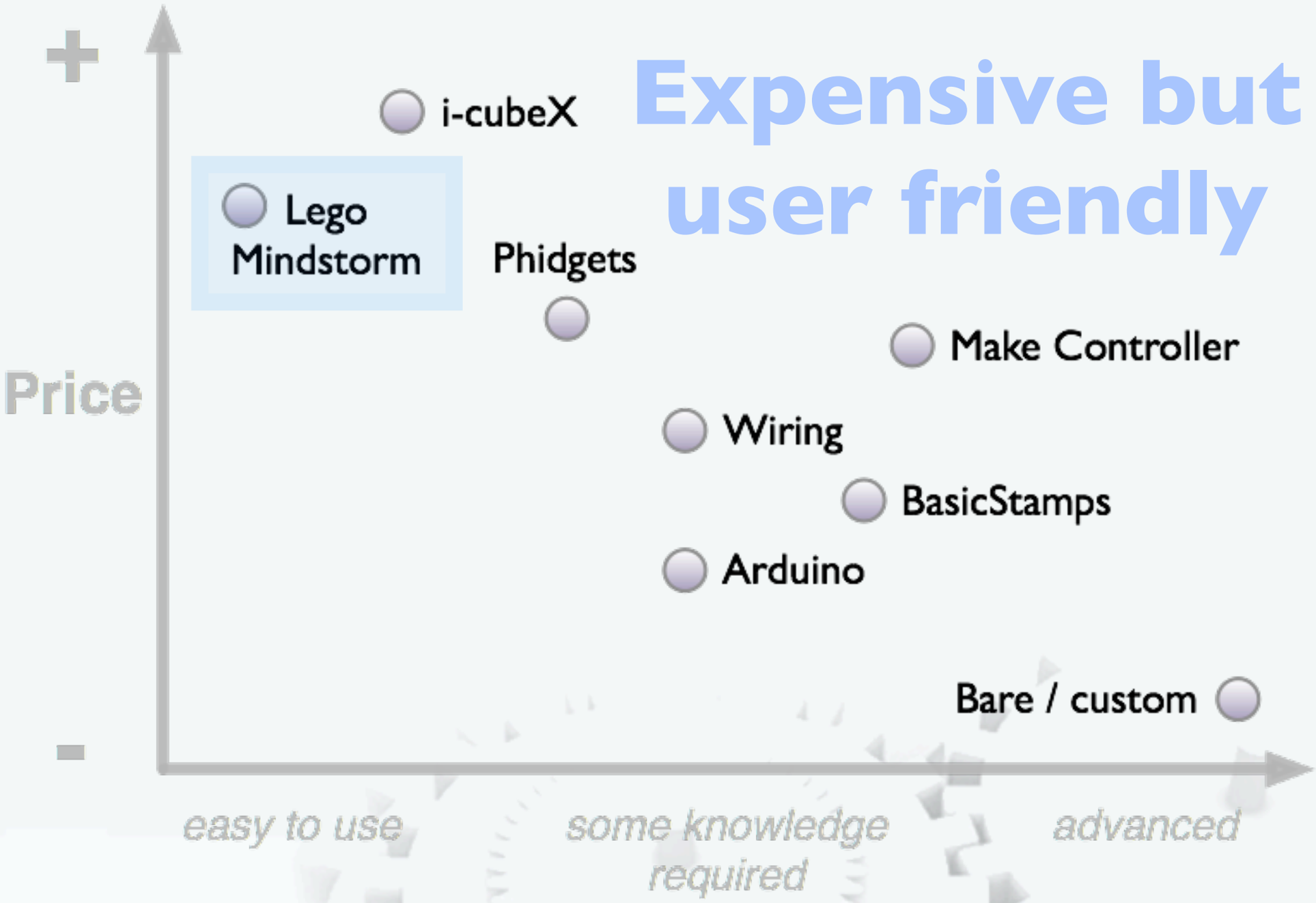
# Existing Toolkits





# Existing Toolkits






**Expensive but user friendly**

# Existing Toolkits

interaction design centre

**Expensive but user friendly**

**Sufficient for almost all needs**

# Existing Toolkits

**Expensive but user friendly**

**Chips and PCBs**

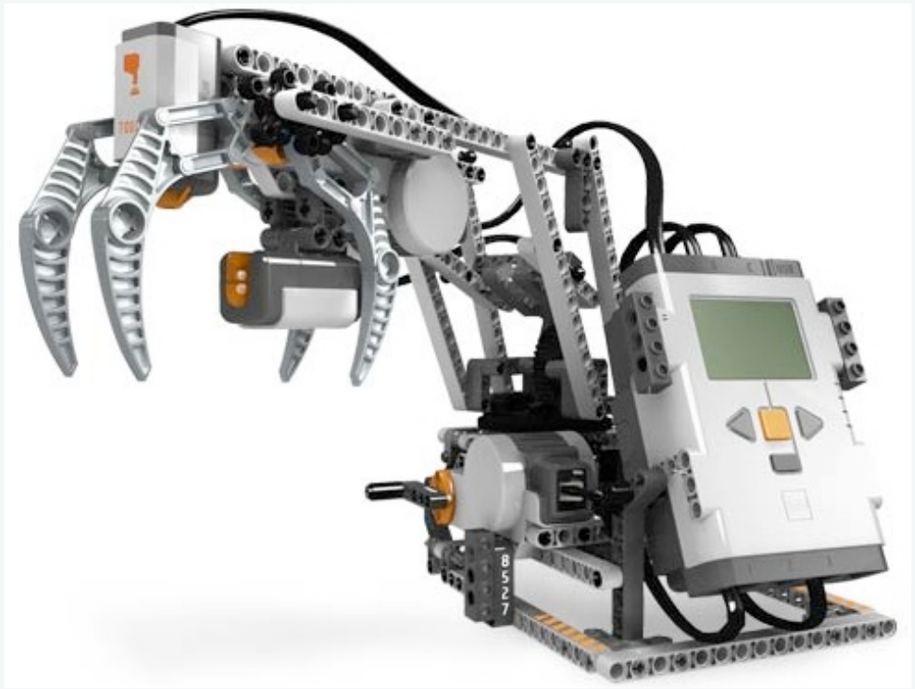
**Sufficient for almost all needs**

# Cost / Difficulty Tradeoff

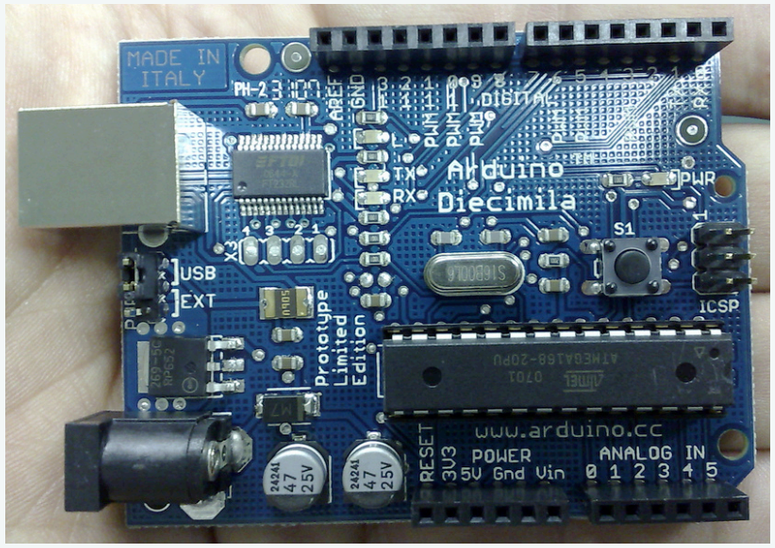
interaction design centre

irish open source technology conference

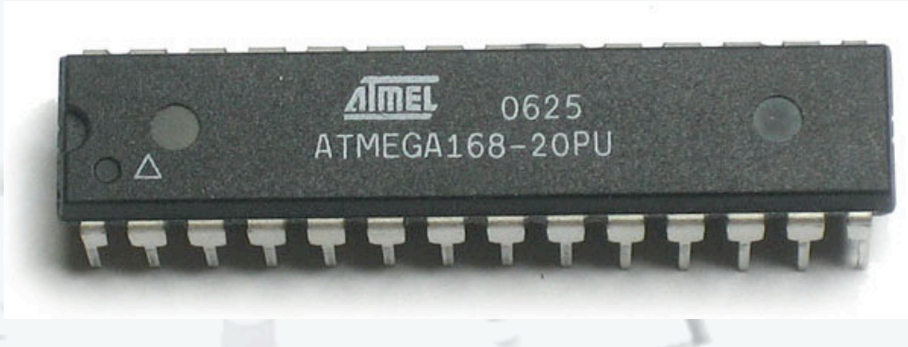
## ☆ Lego Mindstorm NXT



## ☆ Arduino




## ☆ ATMega168

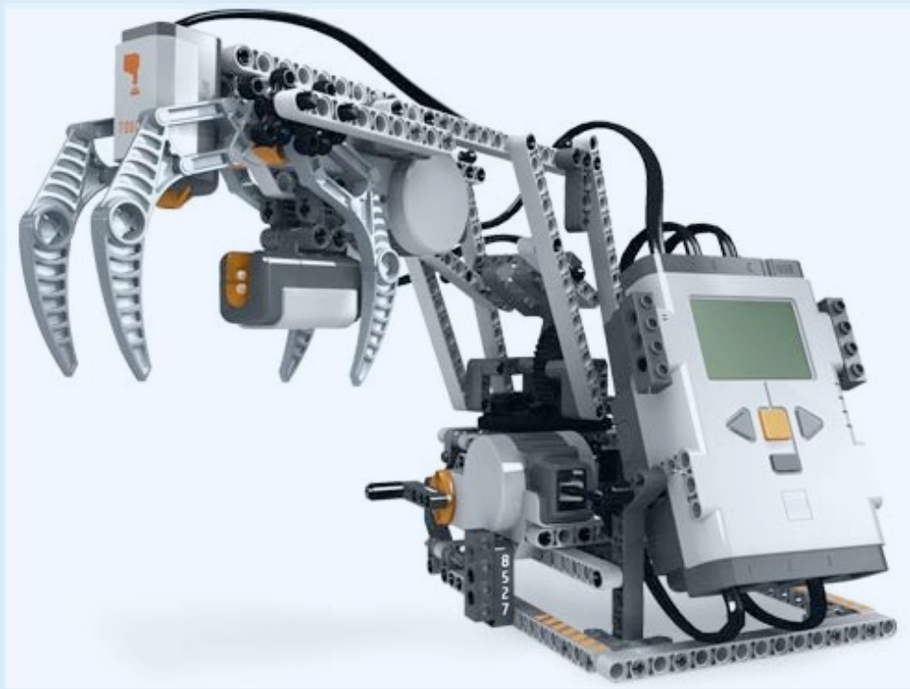


# Cost / Difficulty Tradeoff

interaction design centre

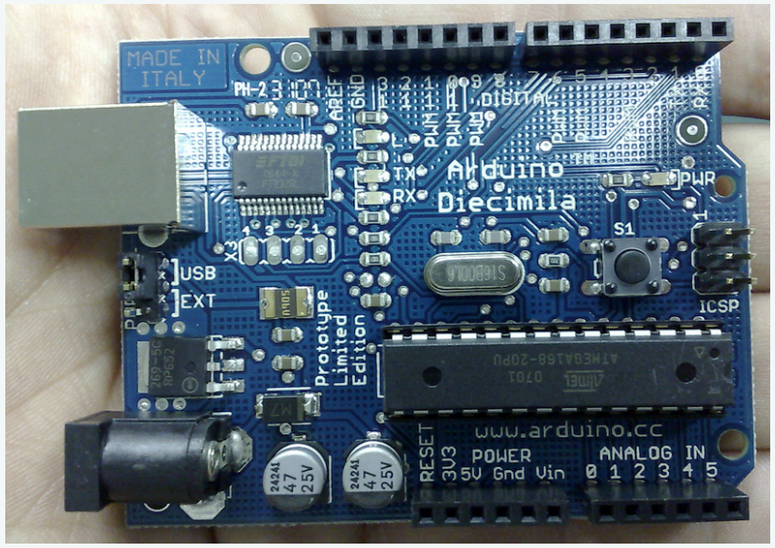
irish open source technology conference

## ☆ Lego Mindstorm NXT

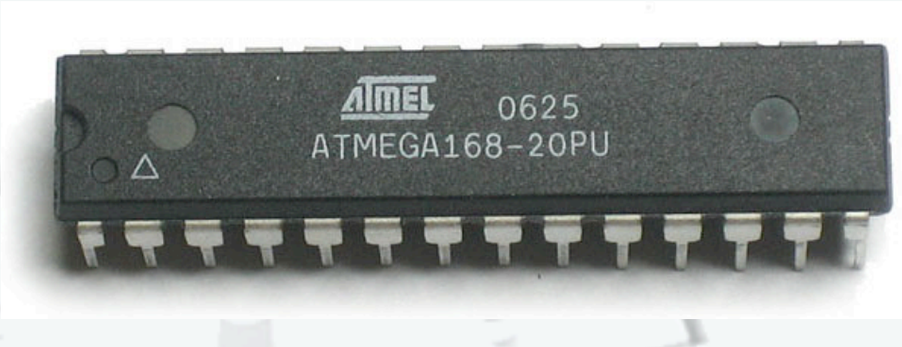


**Approx. ~€250**

## ☆ Arduino




## ☆ ATMega168

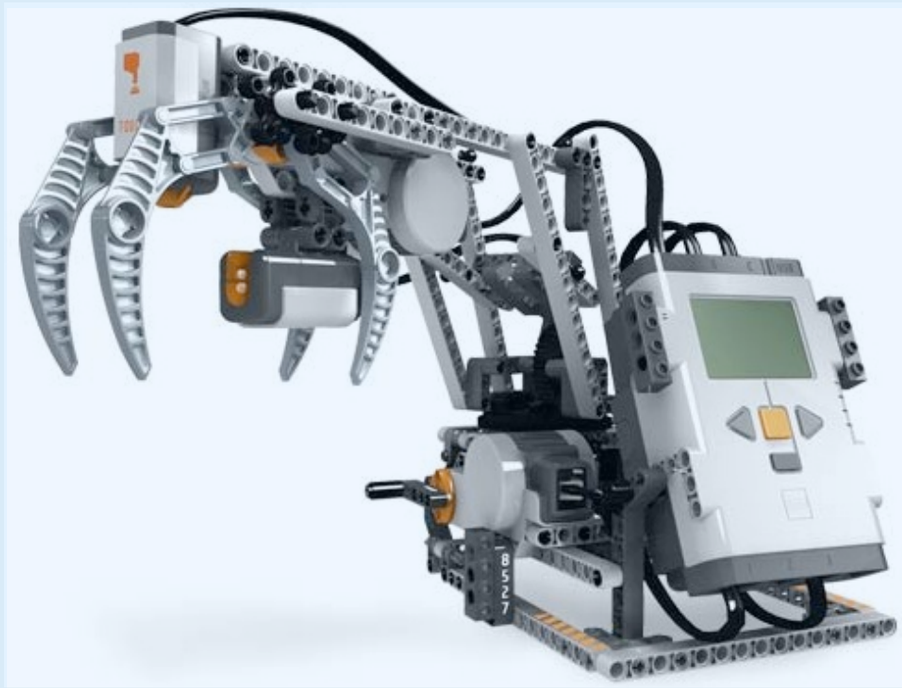


# Cost / Difficulty Tradeoff

interaction design centre

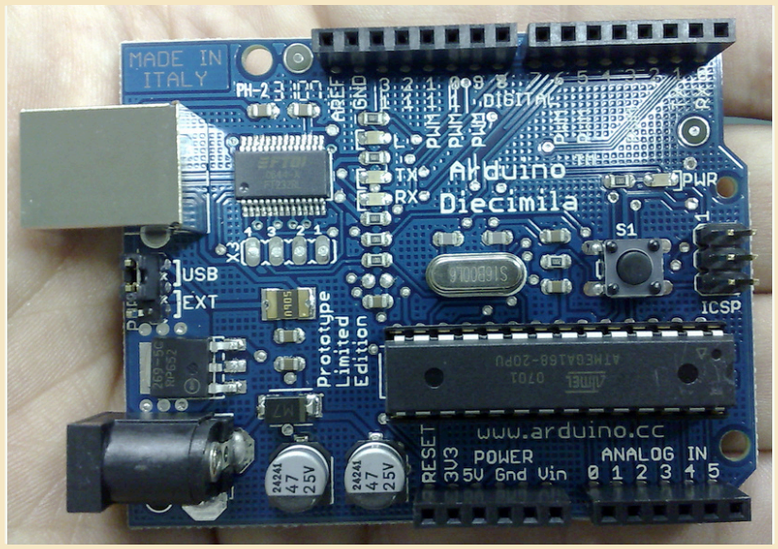
irish open source technology conference

## ☆ Lego Mindstorm NXT



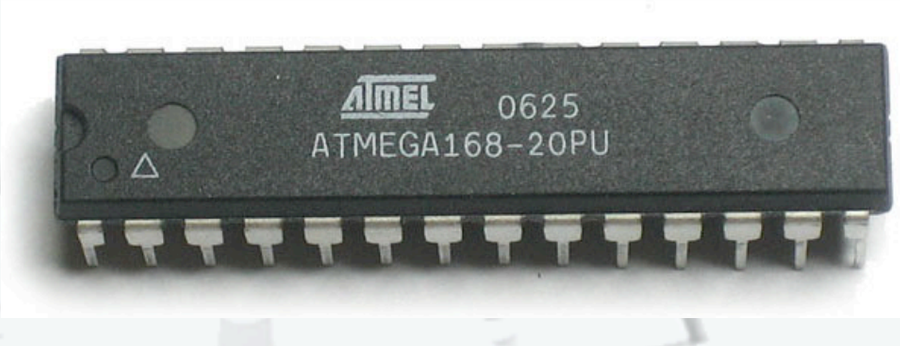
**Approx. ~€250**

## ☆ Arduino




**Approx. ~€25**

## ☆ ATMega168

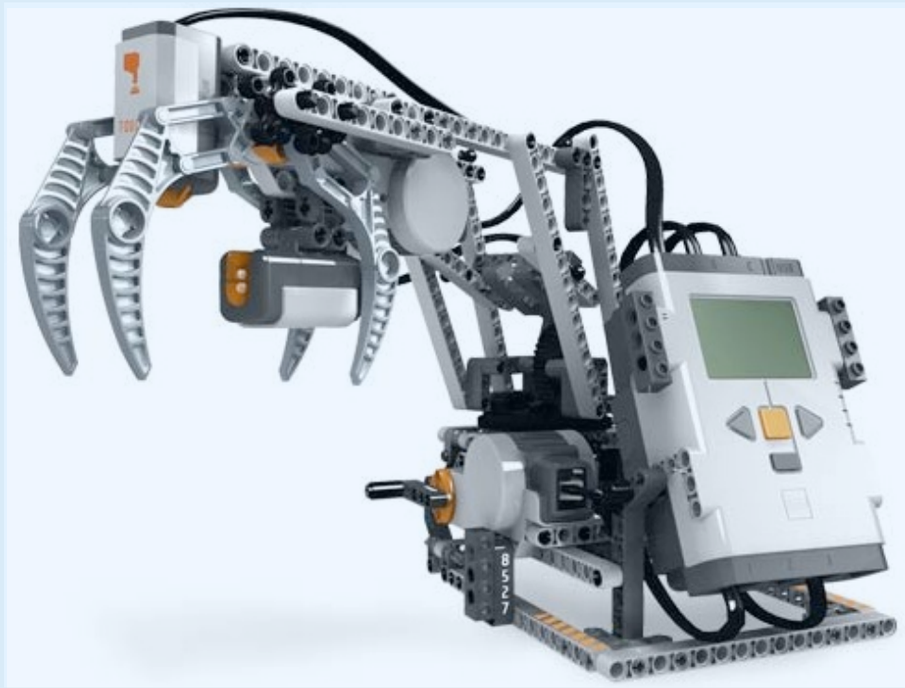


# Cost / Difficulty Tradeoff

interaction design centre

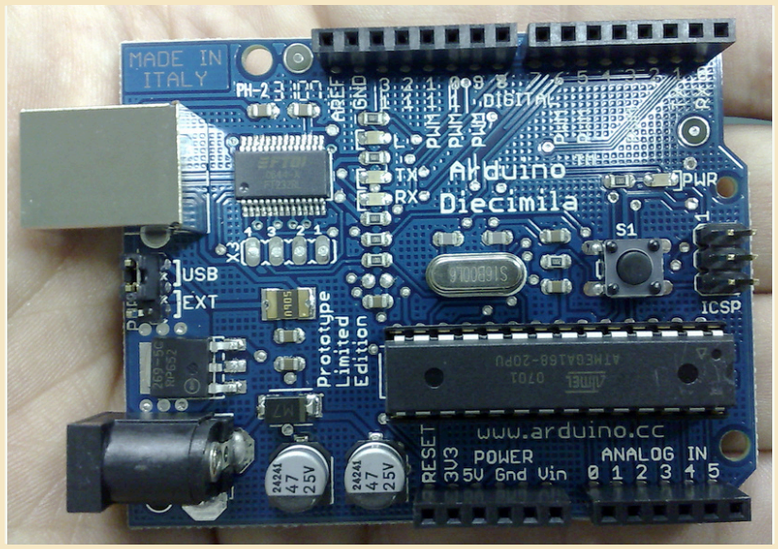
irish open source technology conference

## ☆ Lego Mindstorm NXT



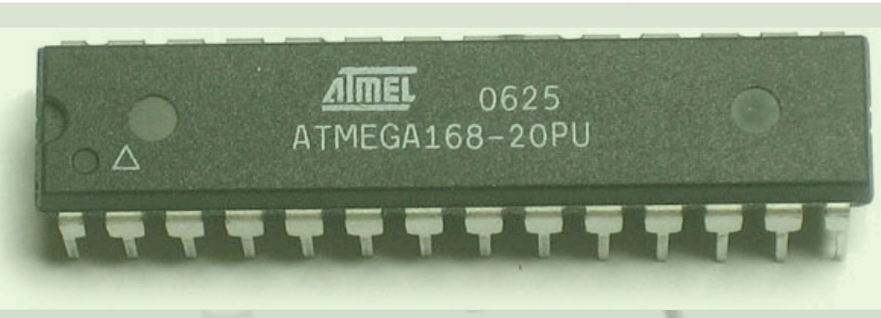
**Approx. ~€250**

## ☆ Arduino



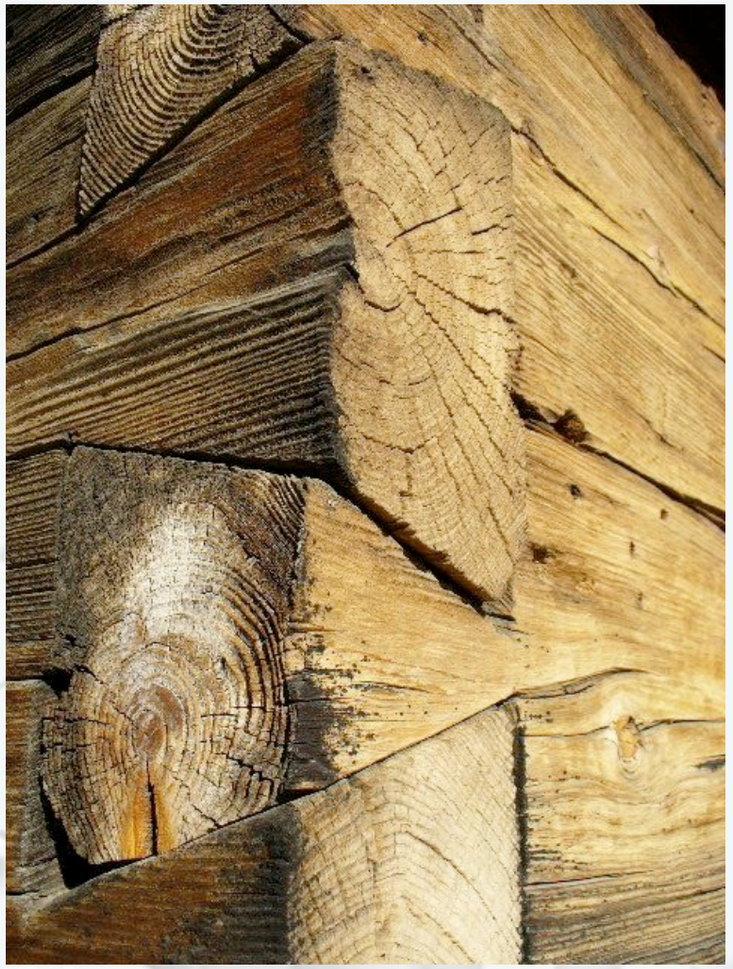

**Approx. ~€25**

## ☆ ATMega168

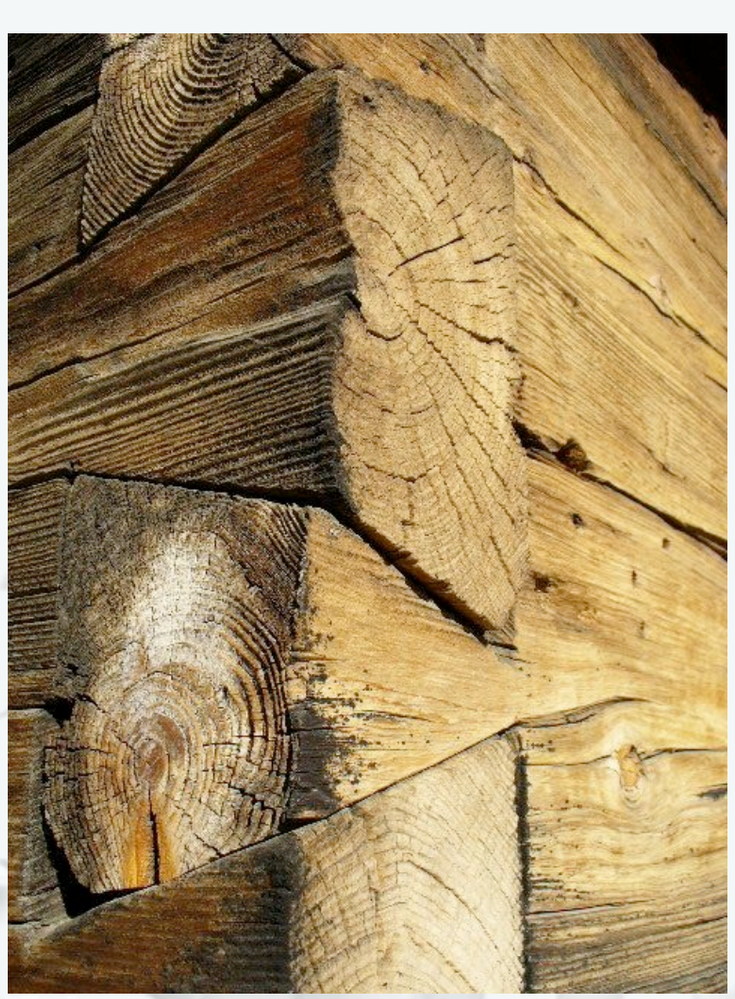



**Approx. ~€4**

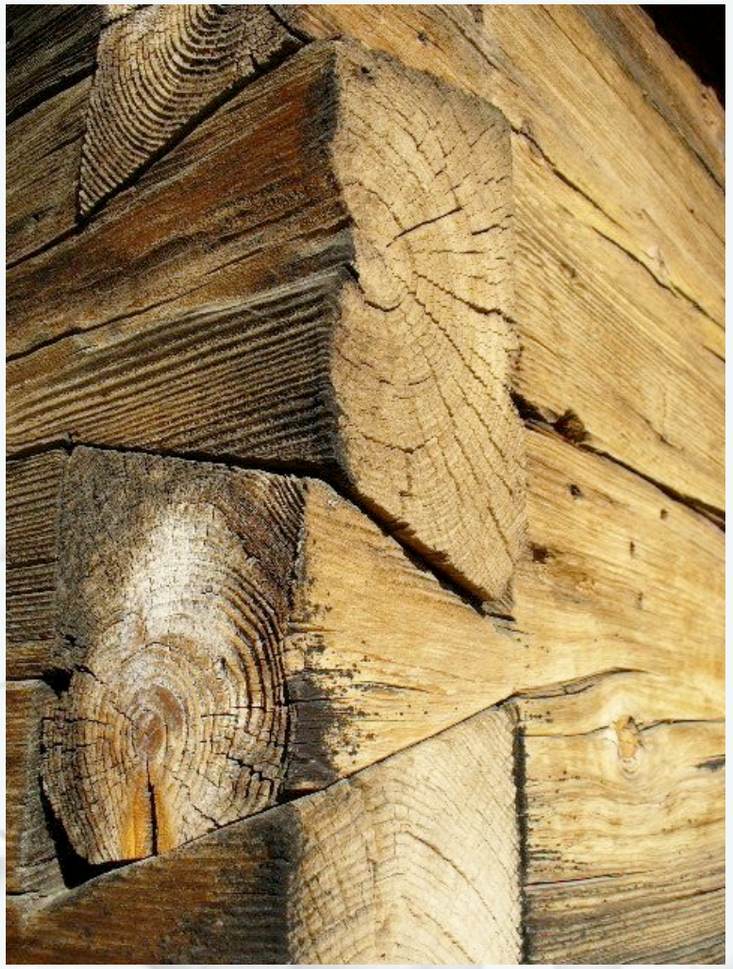
# Opportunistic Development



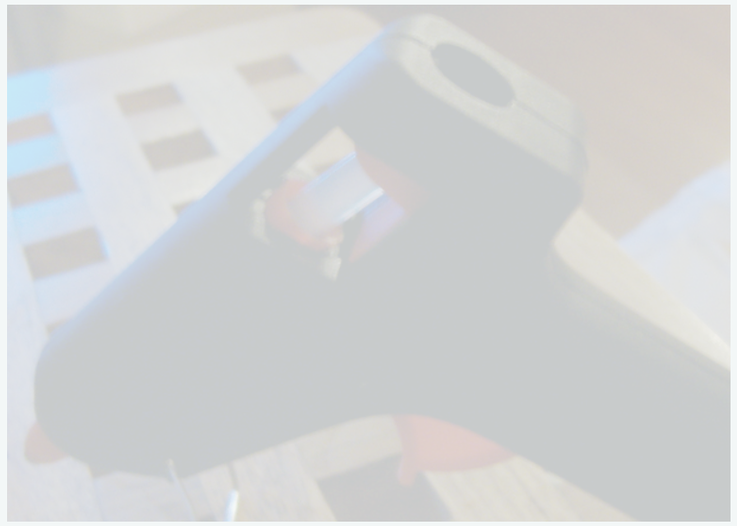
# Big / Heavyweight Software



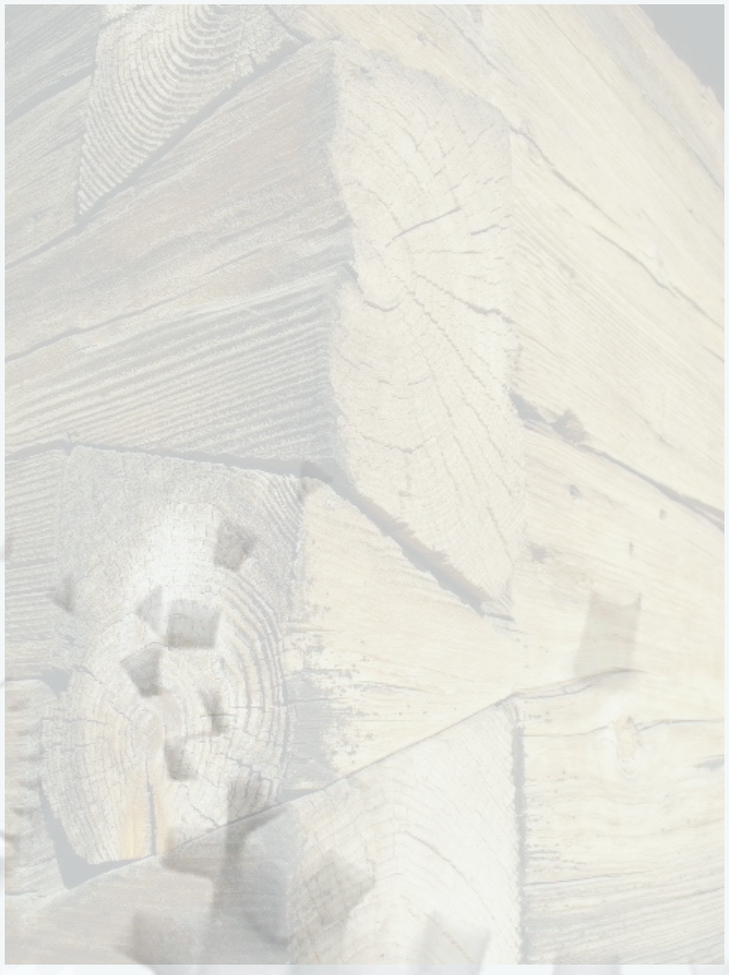
# Glue / Surface Level Integration



# Dovetails / Tight Integration



**Definition**: A *Mash-up* is a combination of existing technologies glued together to create new functionality

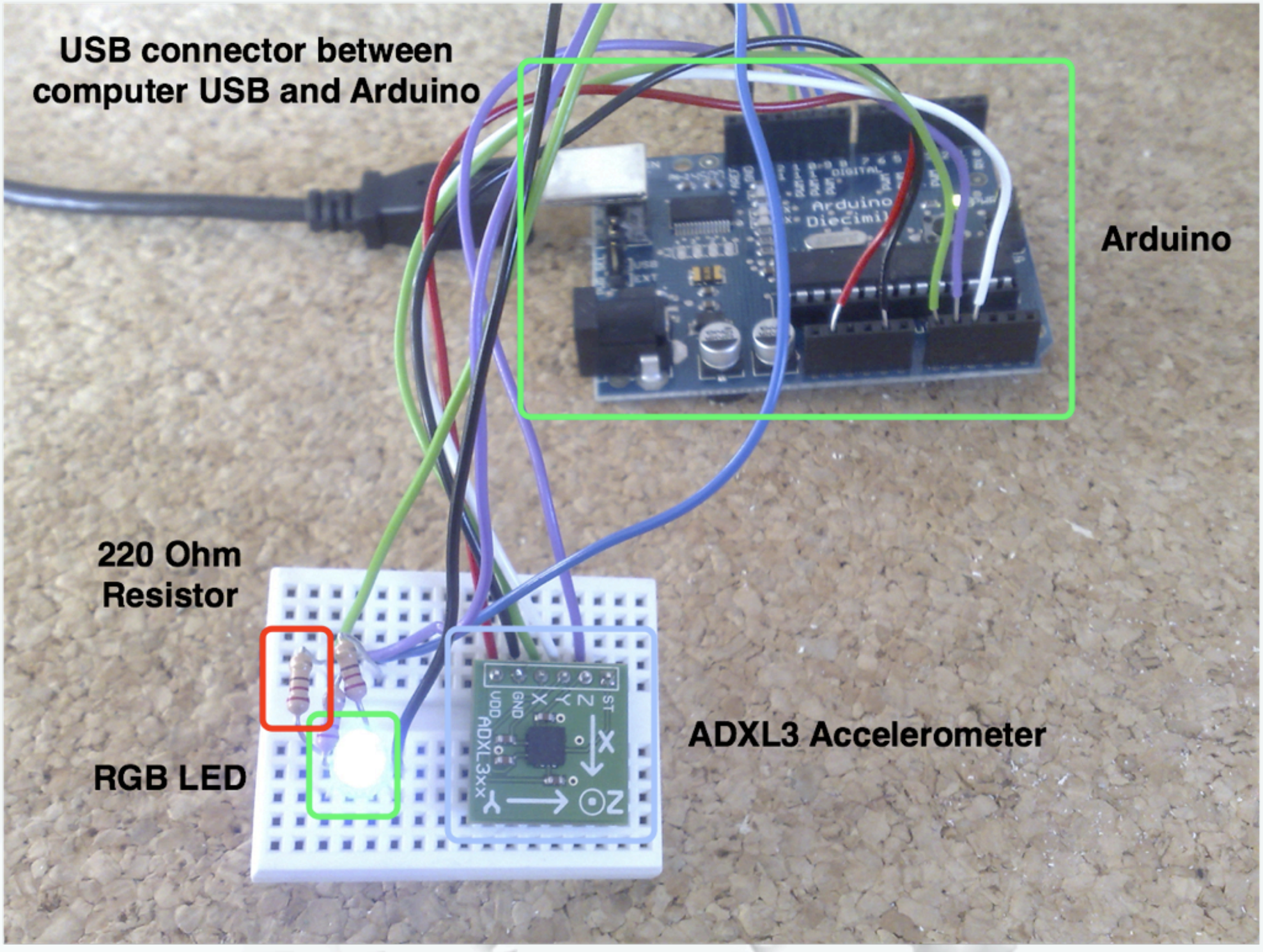


# Examples

# Accelerometer & RGB LED

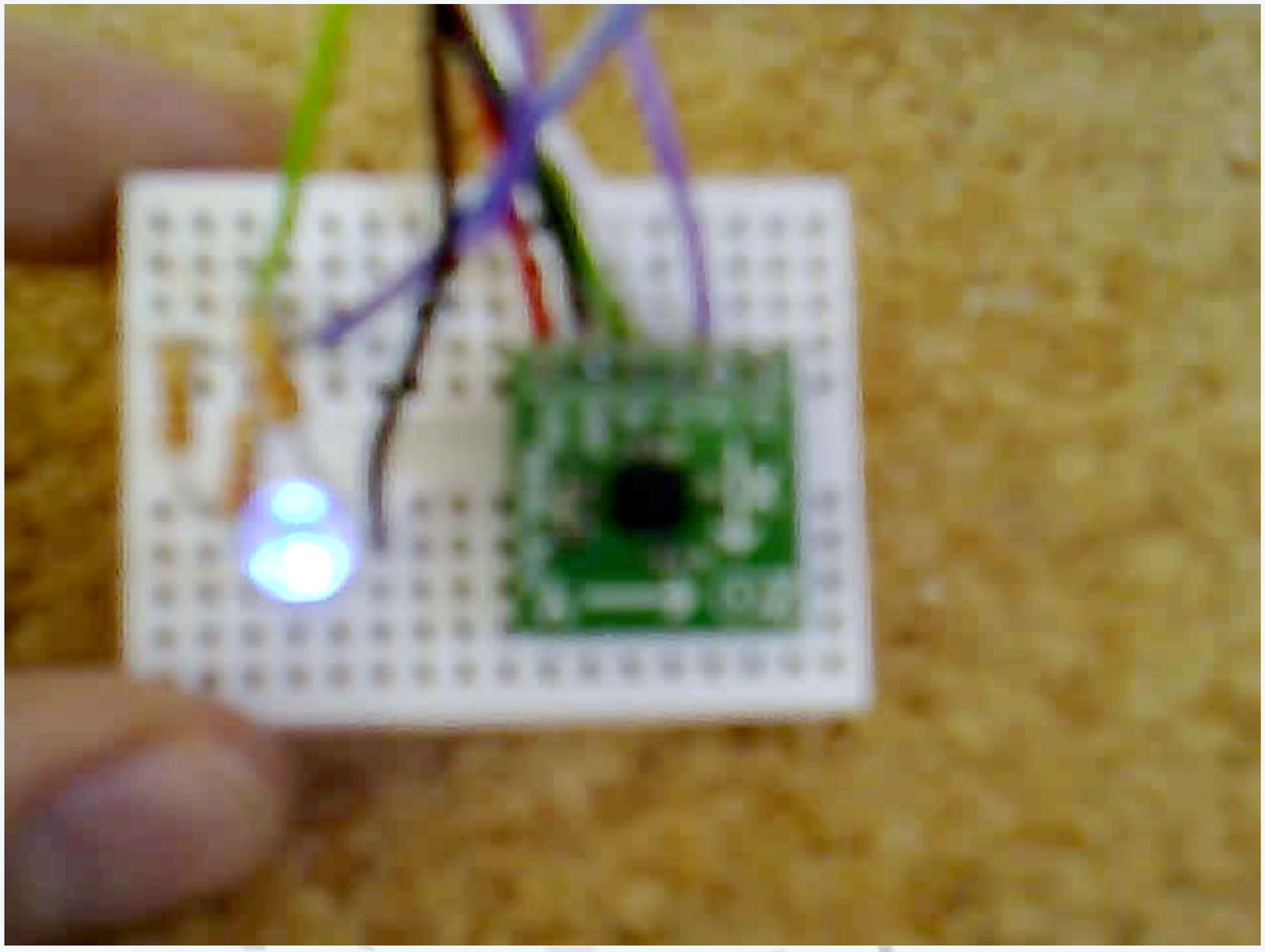
interaction design centre





# Accelerometer & RGB LED

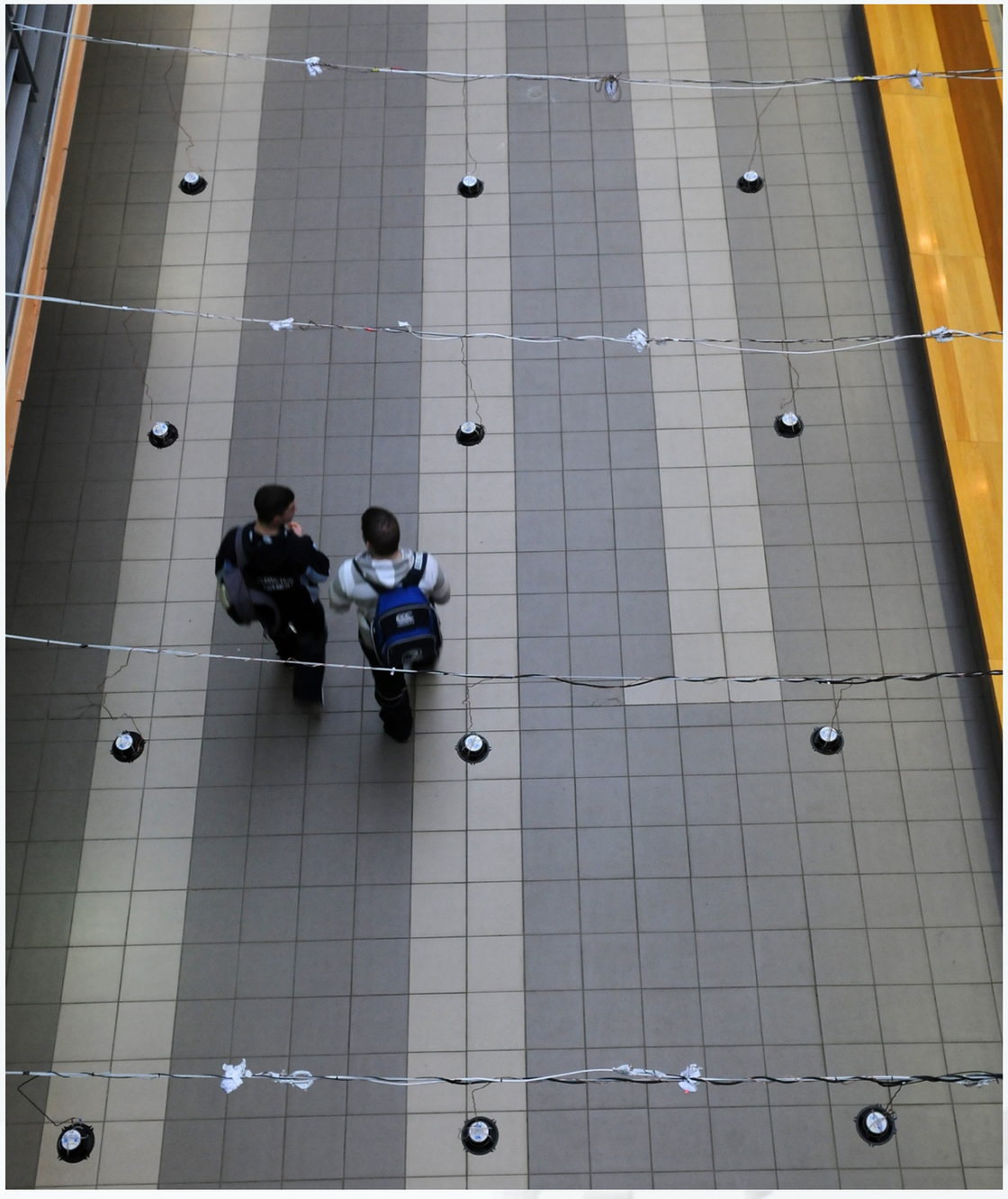




# Hanging Gardens - Another Example

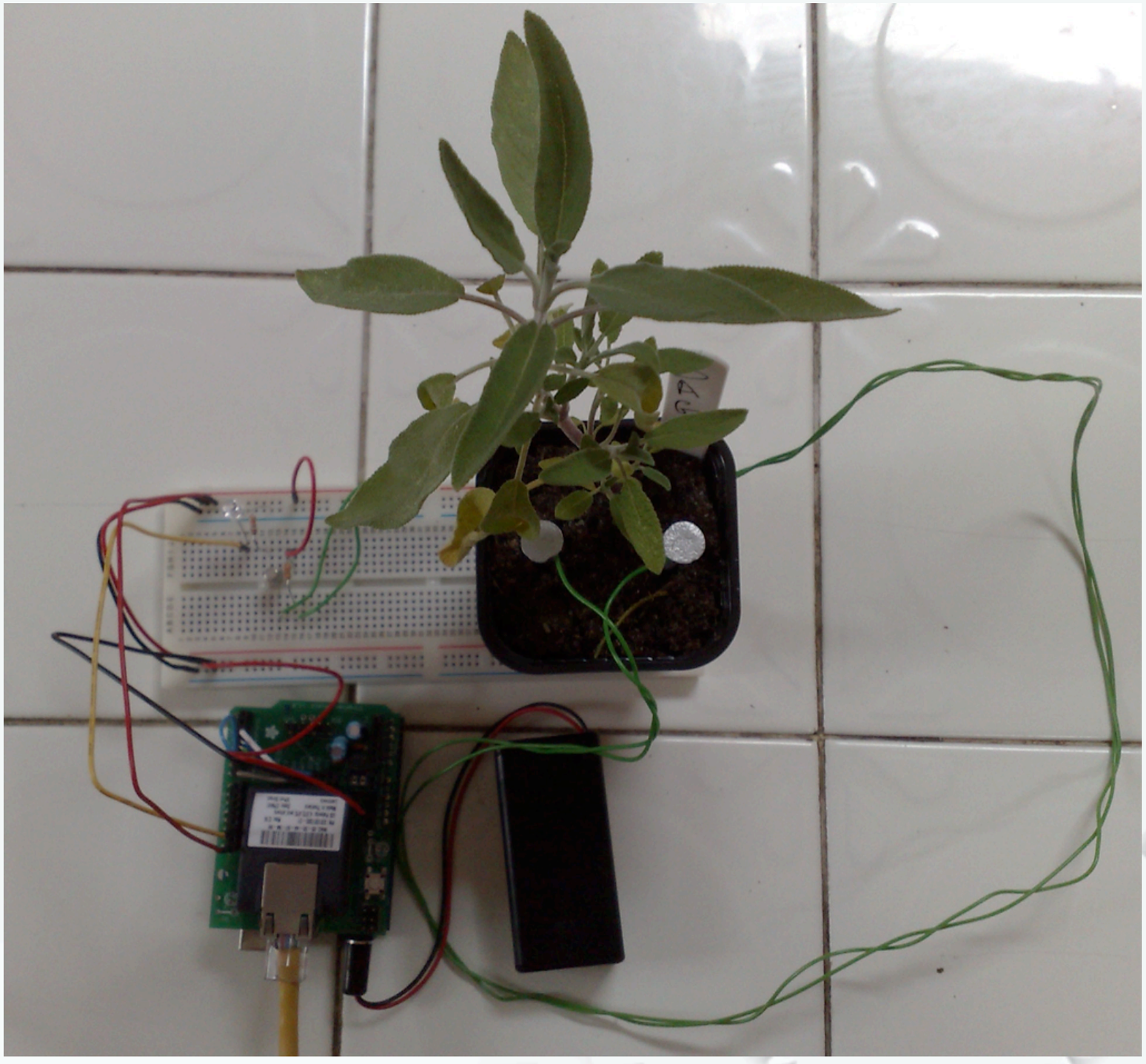
# Hanging Gardens - Another Example





## *Hanging Gardens:*

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



- *Botanicalls*
- Sensors to Arduino
- Arduino to XPort to Twitter

twitter

Name or location search or Login / Join Twitter!

Select Language ...

Hey there! **sageplant1** is using Twitter.

Twitter is a free service that lets you keep in touch with people using the web, your phone, or IM. **Join today** to start receiving **sageplant1's** updates.

**Join today!**

Already using Twitter via SMS or IM? Finish signing up.

# sageplant1

## URGENT! Water me!

2 days ago from web

You didn't water me enough 09:57 PM June 15, 2008 from web

URGENT! Water me! 09:41 PM June 15, 2008 from web

You didn't water me enough 11:13 PM June 14, 2008 from web

RSS Older »

**About**

***Name*** *sageplant1*
***Location*** *Limerick, Ireland*
***Web*** *http://braz.blogs...*
***Bio*** *I'm a Sage Plant that twitters.*

**Stats**

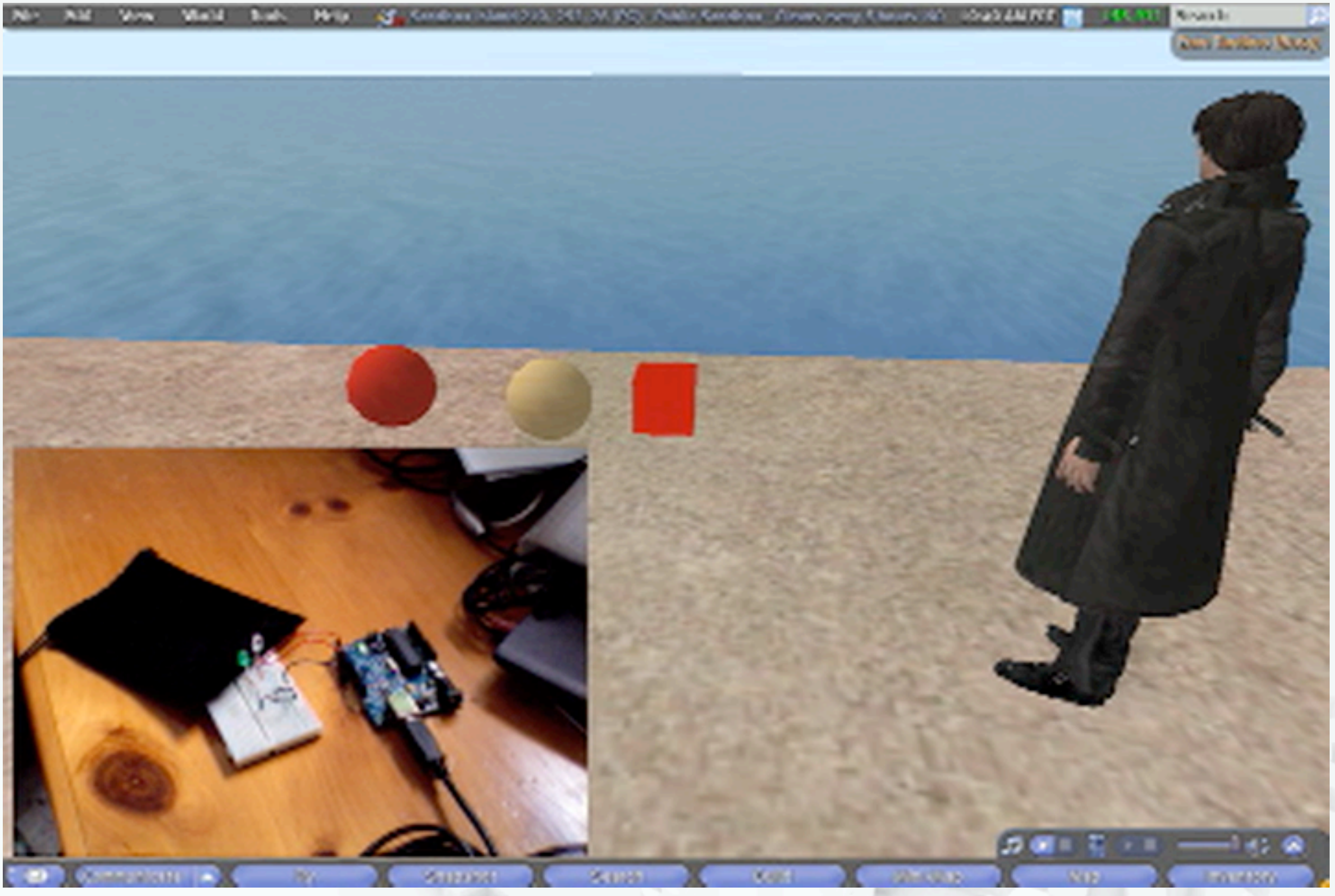
| | |
|---|---|
| Following | 0 |
| Followers | 0 |
| Favorites | 0 |
| Updates | 4 |

© 2008 Twitter About Us Contact Blog Downloads API Help Jobs TOS Privacy

# Example of SL to RL

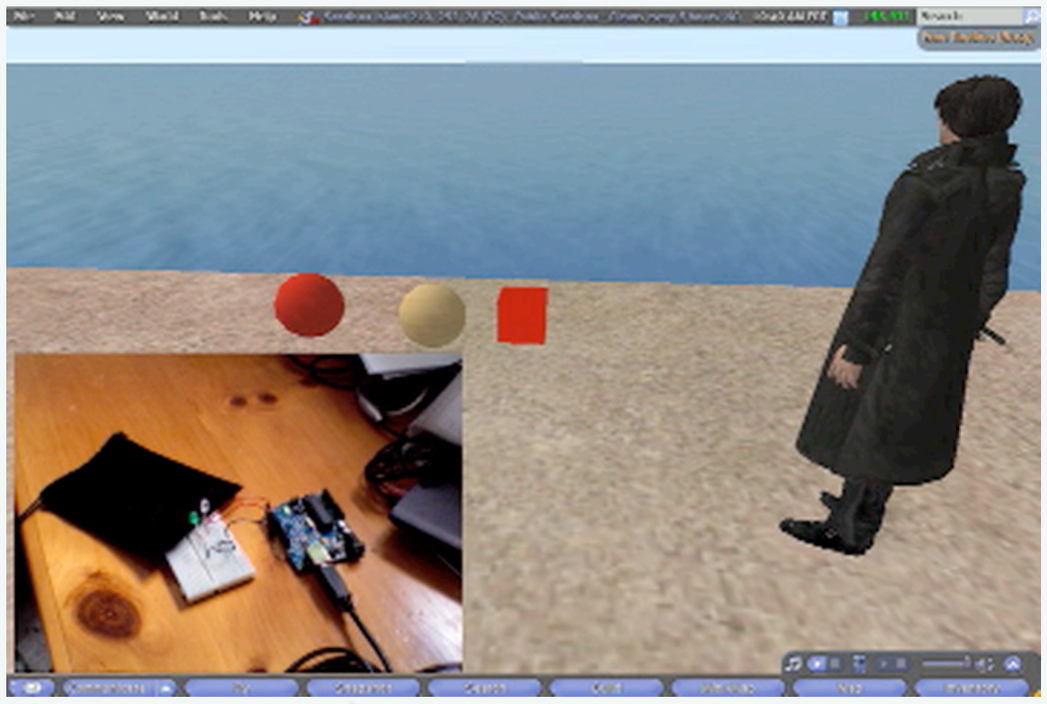
interaction design centre





# *SL to RL*

- LSL script for SL objects
- LSL to PHP webserver with connected Arduino
- PHP to Arduino's serial port



# Spimes - An Internet of Things

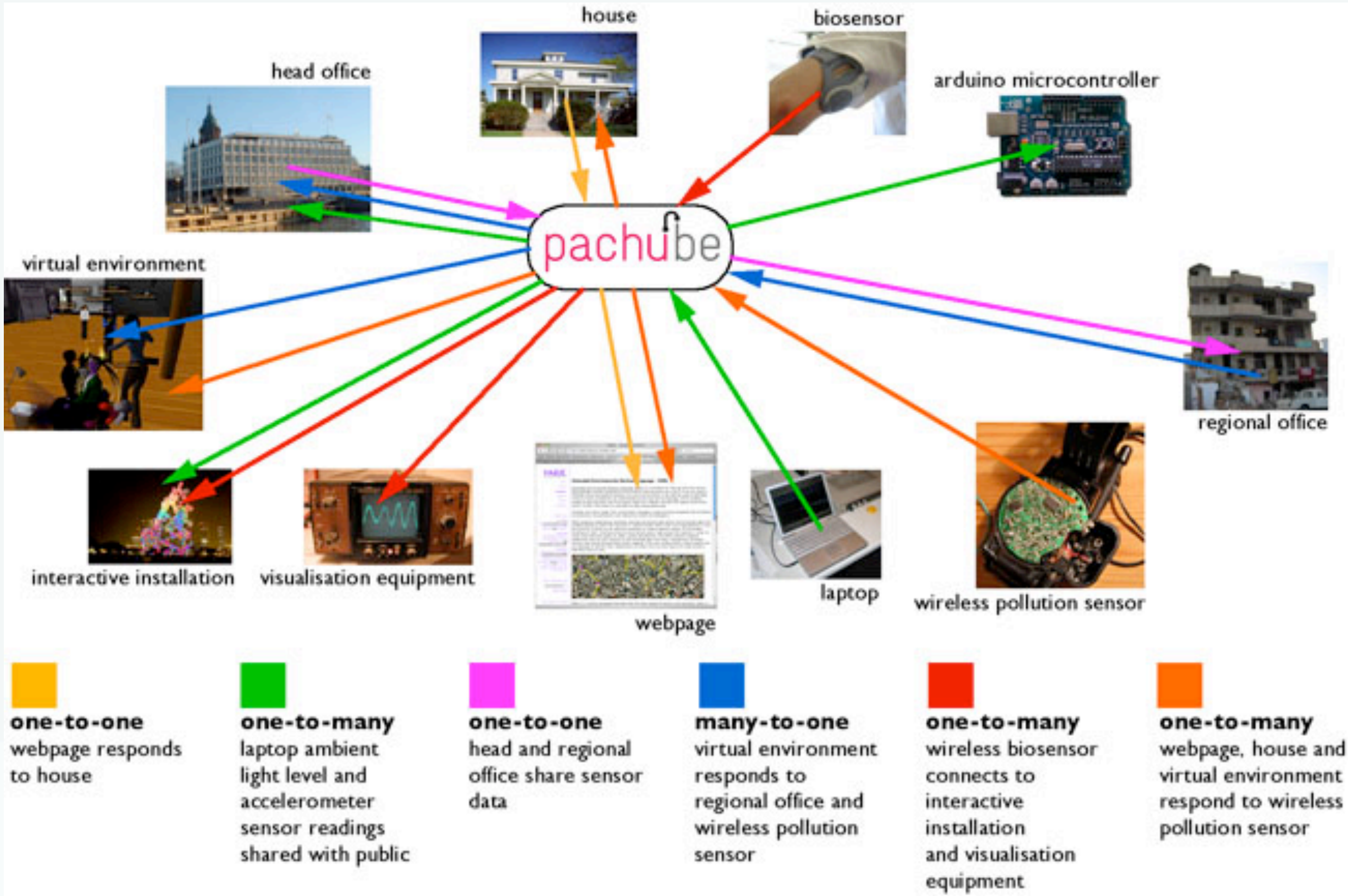
interaction design centre

irish open source technology conference

# Spimes - An Internet of Things

interaction design centre

irish open source technology conference




**one-to-one**
webpage responds to house

**one-to-many**
laptop ambient light level and accelerometer sensor readings shared with public

**one-to-one**
head and regional office share sensor data

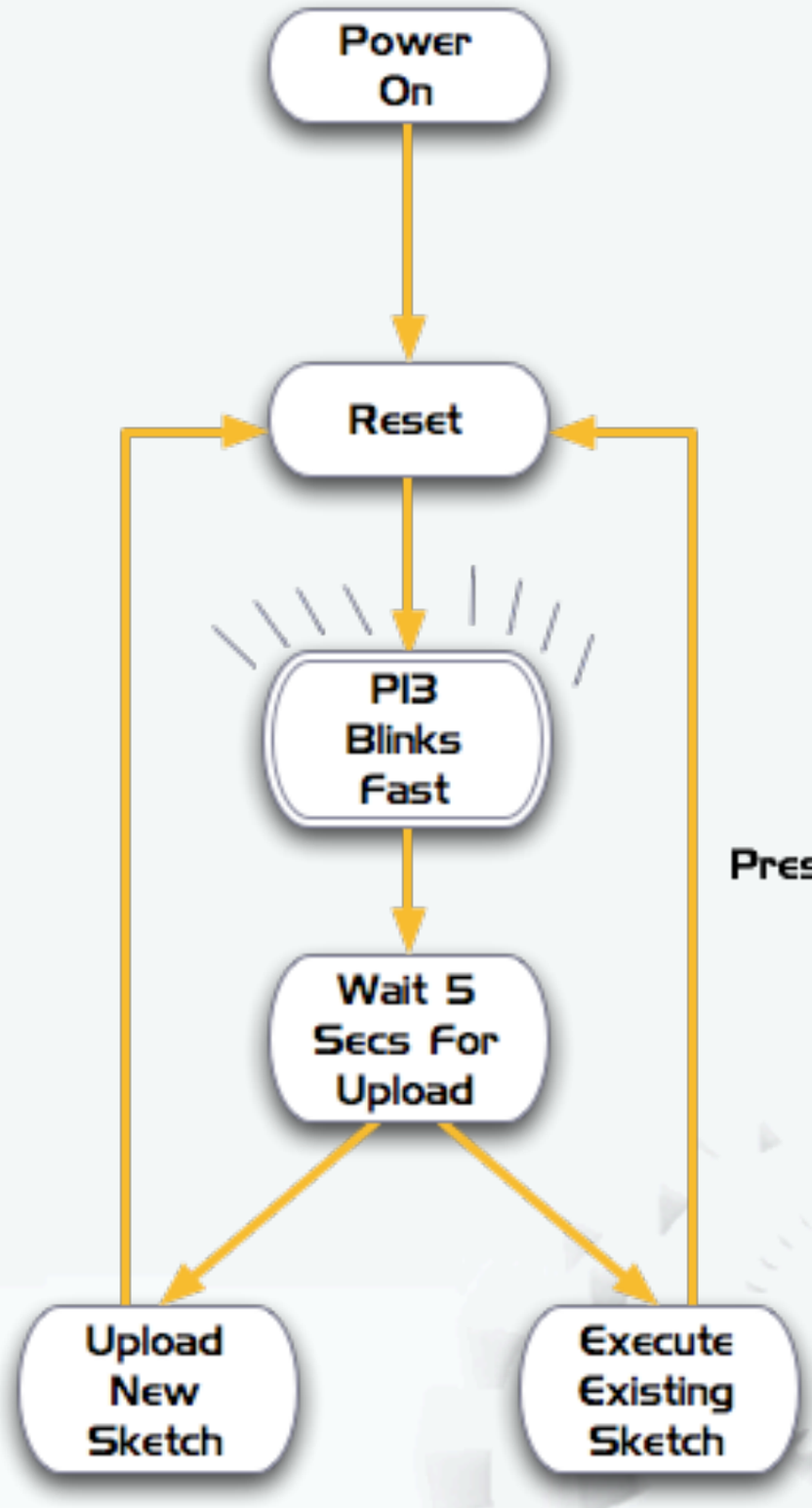
**many-to-one**
virtual environment responds to regional office and wireless pollution sensor

**one-to-many**
wireless biosensor connects to interactive installation and visualisation equipment

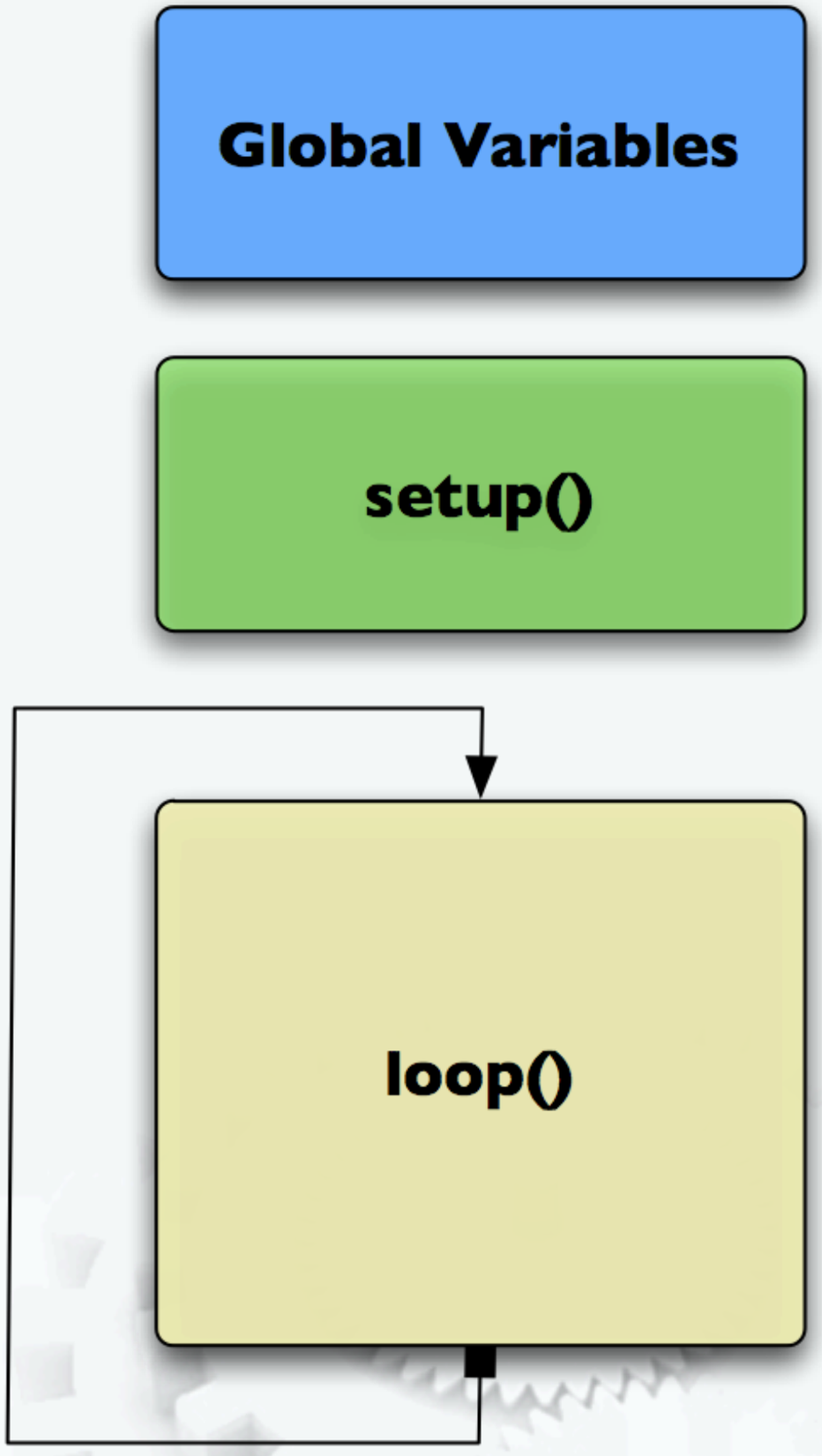
**one-to-many**
webpage, house and virtual environment respond to wireless pollution sensor

# Programming

# Programming an Arduino



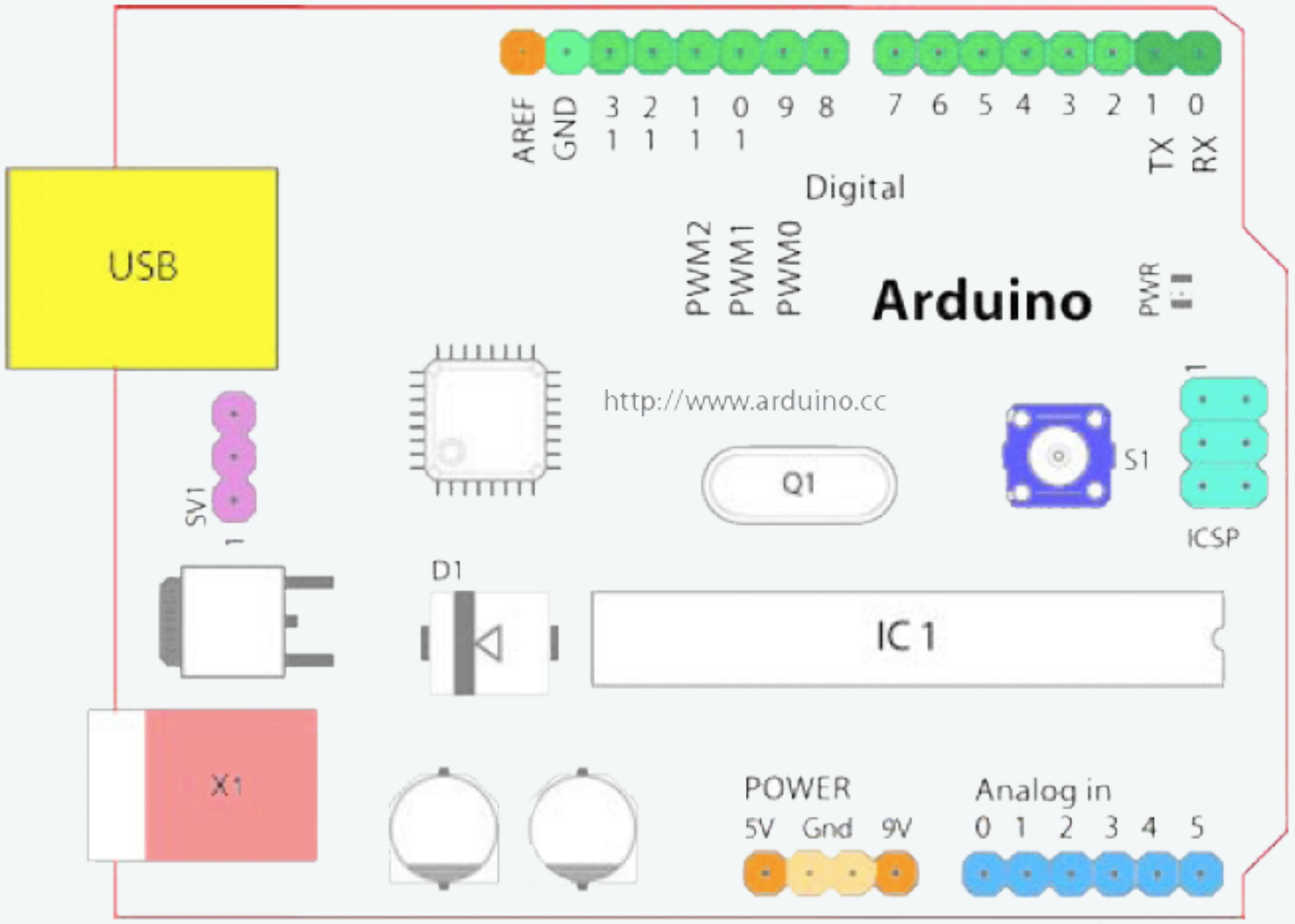

- Write program
- Compile (check for errors)
- Reset board
- Upload to board


Global Variables
setup()
loop()

- pinMode()
- digitalWrite() / digitalRead()
- analogRead() / analogWrite()
- delay()
- millis()

# Input / Output






- 14 *Digital IO* (pins *0 - 13*)
- 6 *Analog In* (pins *0 - 5*)
- 6 *Analog Out* (pins 3, 5, 6, *9, 10, 11*)

# 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 arduino.cc
- 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

# Blinking LED



```
/* 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

```
/* 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
        }
}
```

Initialise some of the variables

# Blinking LED

```
/* 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
        }
}
```

Setup LED pin and serial connection

# Blinking LED

```
/* 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
        }
}
```

Loop - Reading the serial for info, when something is received turn the LED on

# Push button LED



```
/* 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
        }
}
```

# Push button LED



```
/* 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
        }
}
```

Initialise some of the variables

# Push button LED



```
/* 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
```

Setup LED pin, switch pin and serial connection

```
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
        }
}
```

# Push button LED



```
/* 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
}
```

Loop - Reading the button for info, when button is press turn the LED on and signal the computer of change

```
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
        }
}
```

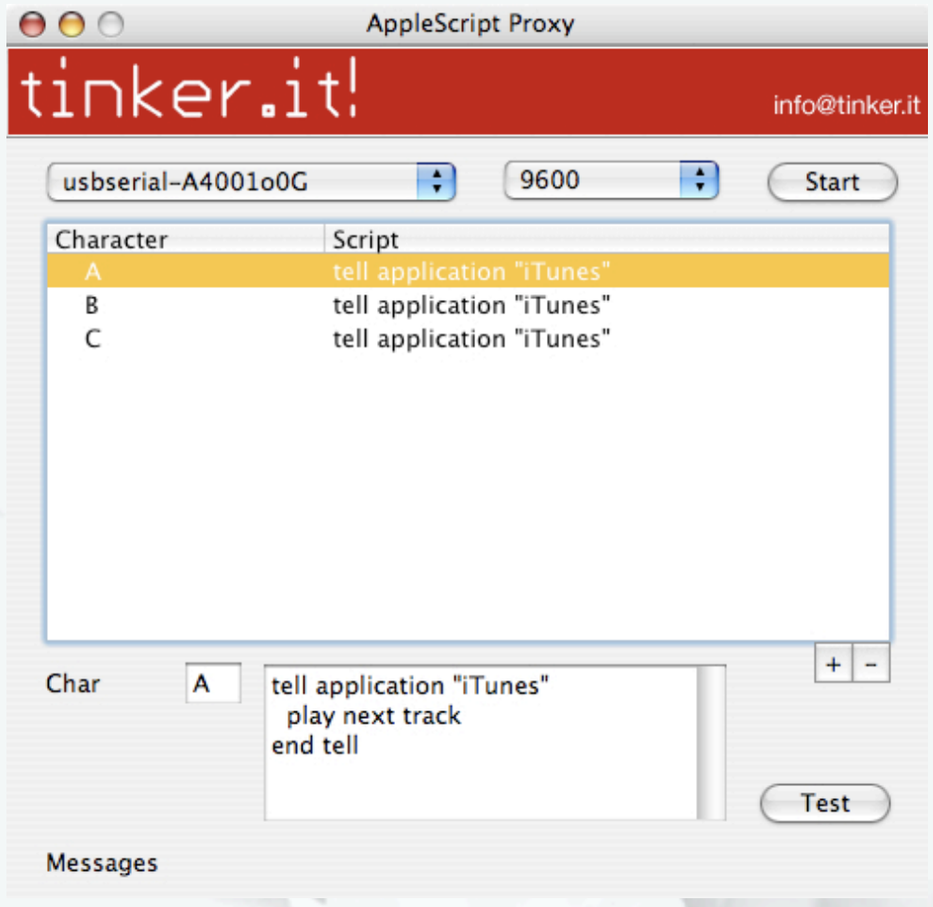
# Useful Stuff

*Proxy:* Conversion of communication to another type

- *Network serial (Serial to TCP)*
- *TinkerProxy / Griffin Proxi*
- *osculator*
- *Girder (Windows)*
- *Shion, Indigo*
- *Sydewynder*

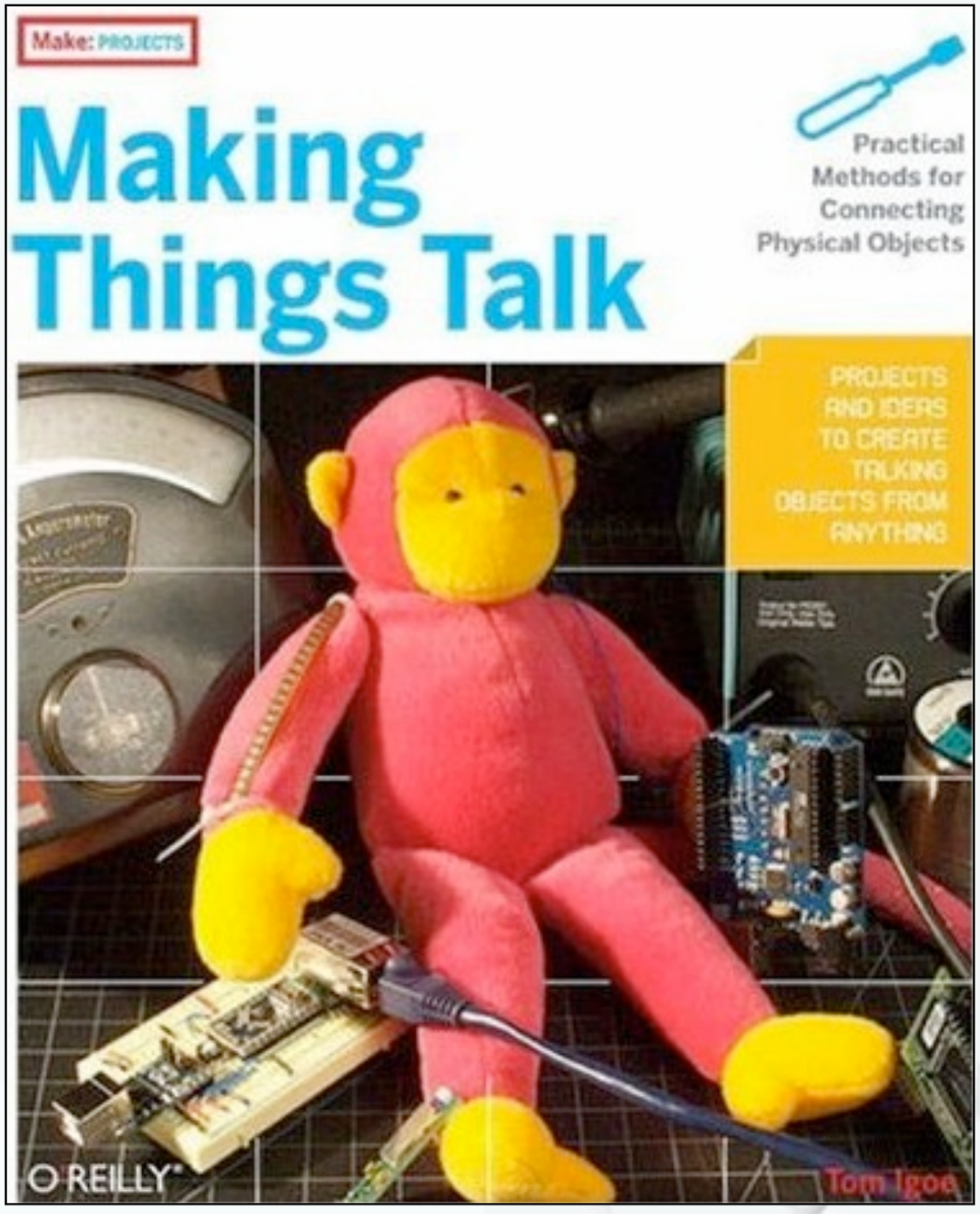
*Protocol:* Structured conversation

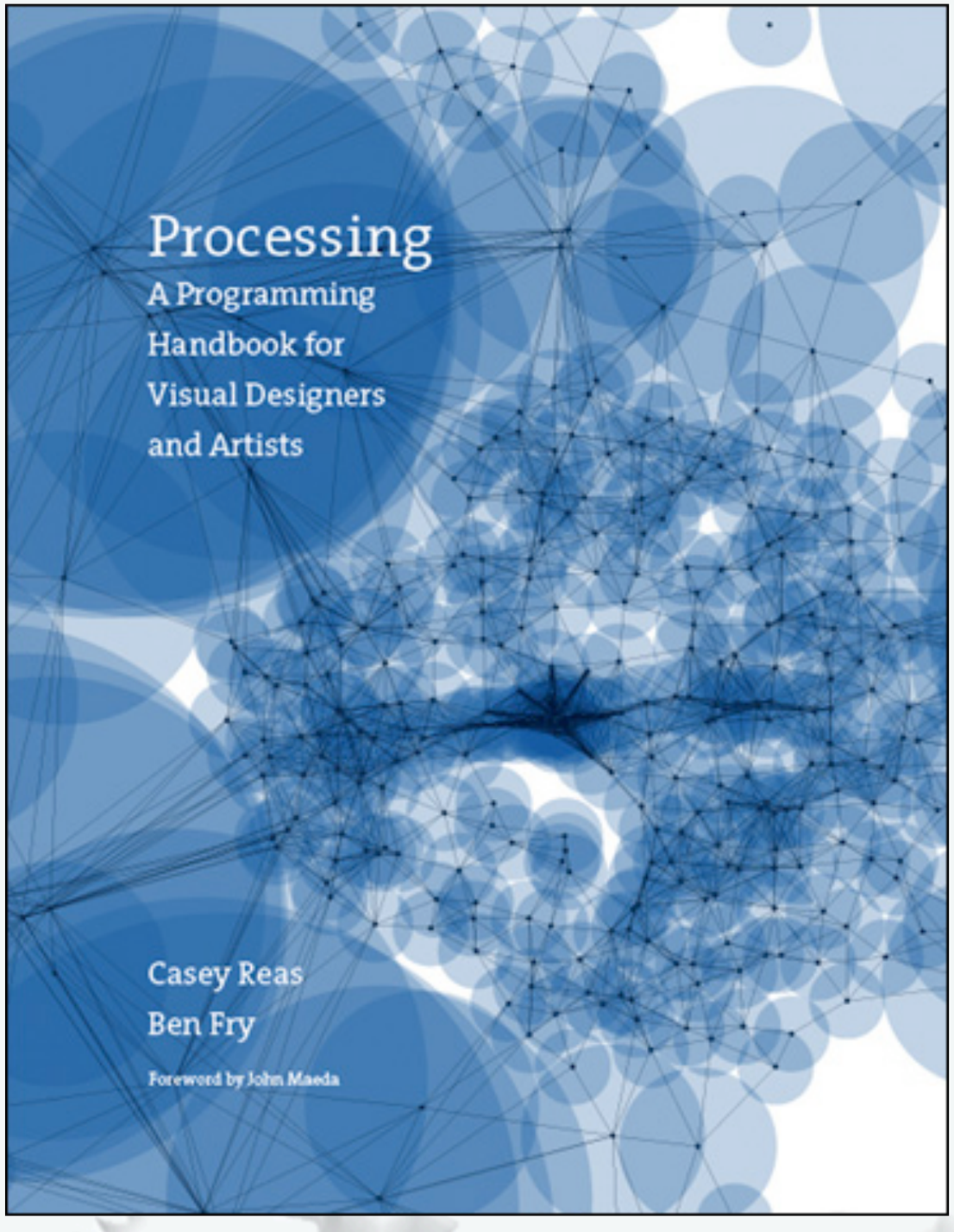
- *Midi / OSC*
- *DMX512*
- *X10, INSTEON*



# Suggested Books





# Useful Links



- *Arduino - http://www.arduino.cc/*
- *Arduino lectures - http://www.slideshare.net/eoinbrazil*
- *Tod E. Kurt's blog (check his Spooky Arduino projects) - http://todbot.com/blog/category/arduino/*
- *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*
- *Usman Haque and Adam Somlai-Fischer - ``Low tech sensors and actuators for artists and architects''*