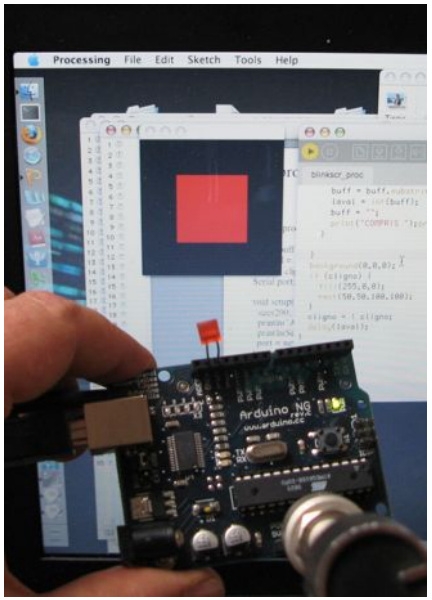
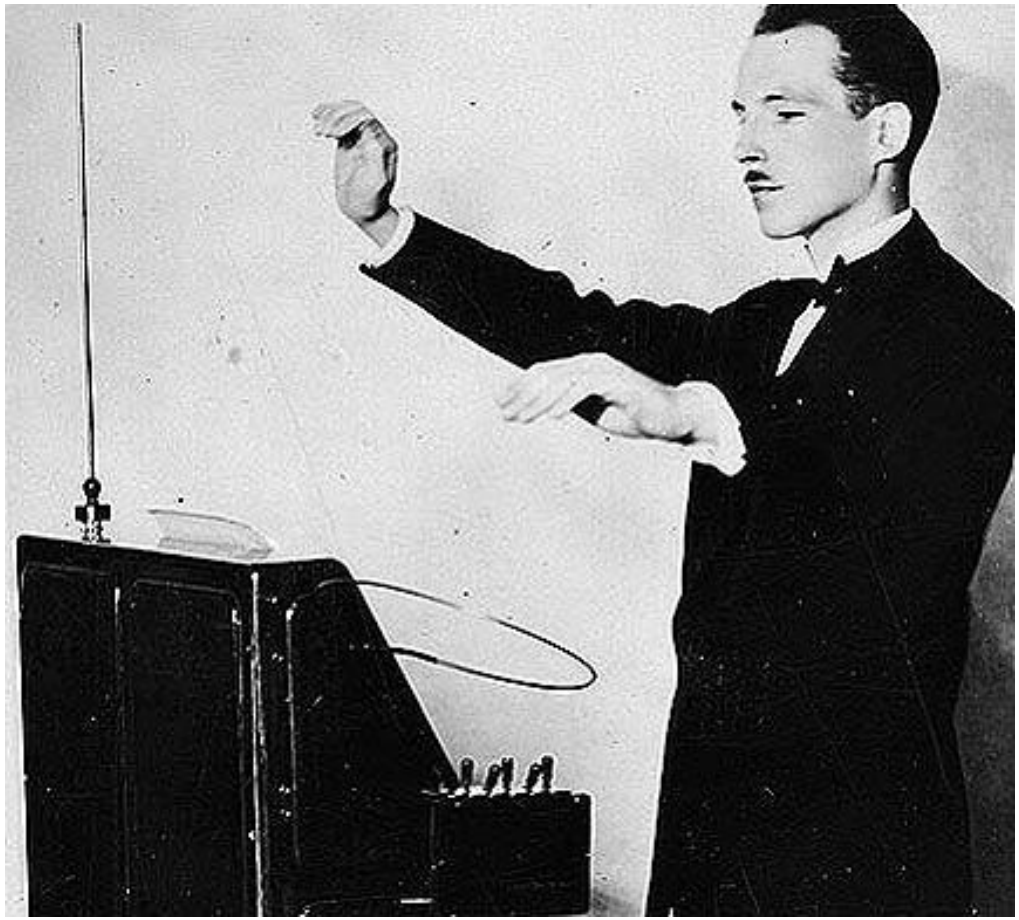
CNAM - NSY116 - janvier 2015

Un peu de physical computing

Pierre Cubaud
cubaud @ cnam.fr



- Smart objects
- Périphériques riches
- Wiimote
- La carte Arduino
- Couplage Arduino/Processing



"Physical computing" ?

Physical computing
From Wikipedia, the free encyclopedia
• *Have questions?*
Find out how to ask questions and get answers. •Jump to: navigation, search

Physical computing, in the broadest sense, means building interactive physical systems by the use of software and hardware that can sense and respond to the analog world. While this definition is broad enough to encompass things such as smart automotive traffic control systems or factory automation processes, it is not commonly used to describe them. In the broad sense, physical computing is a creative framework for understanding human beings' relationship to the digital world. In practical use, the term most often describes handmade art, design or DIY hobby projects that use sensors and microcontrollers to translate analog input to a software system, and/or control electro-mechanical devices such as motors, servos, lighting or other hardware.

(Greenberg, UIST'01)

INTRODUCTION

In the last decade, various movements embraced human-computer interface designs that include physical user interfaces augmented by computing power. These include *ubiquitous computing* and *calm technology* [15], *pervasive computing* [1], *tangible user interfaces* [7], *information appliances* [12] and *context-aware computing* [3].

Researchers in these areas have demonstrated many simple but exciting examples of physical user interfaces. Ishii and

Physical Computing is an approach to learning how humans communicate through computers that starts by considering how humans express themselves physically. In this course, we take the human body as a given, and attempt to design computing applications within the limits of its expression.

(Interactive Telecom. Program ITP NYU)

Les « phidgets » (S. Greenberg, C. Fitchett, U. Calgary, 2001)

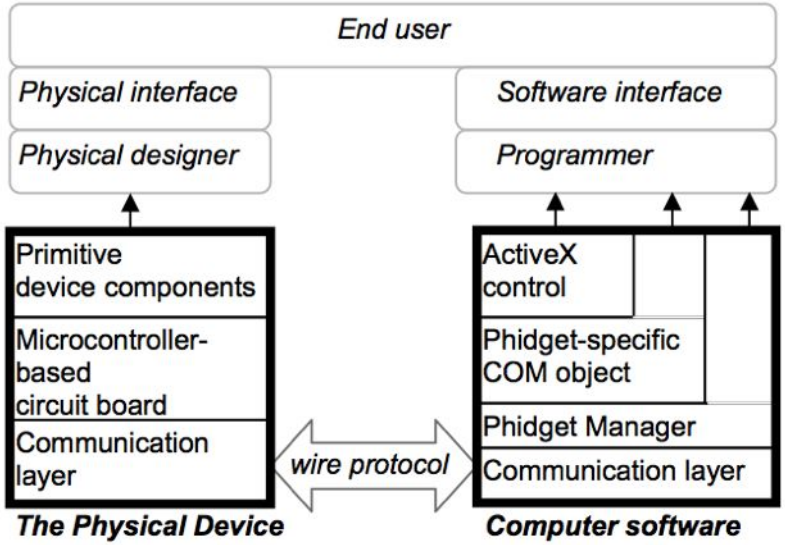
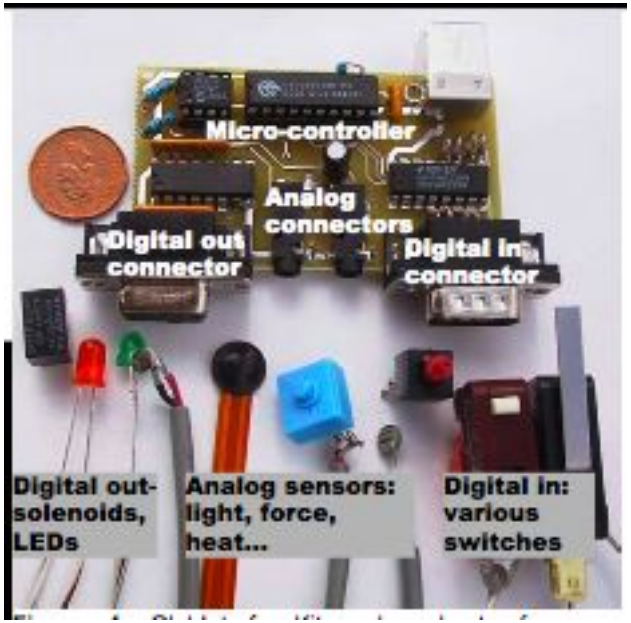
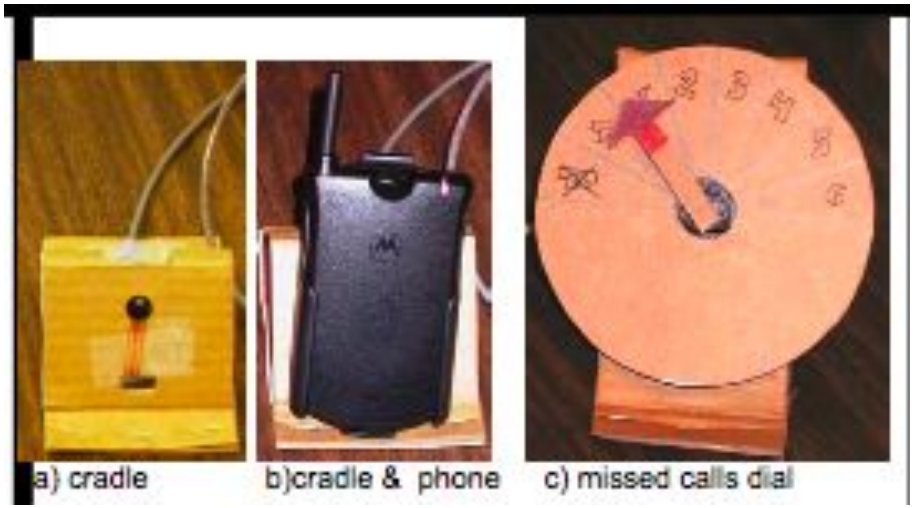


Figure 5. Phidget Architecture





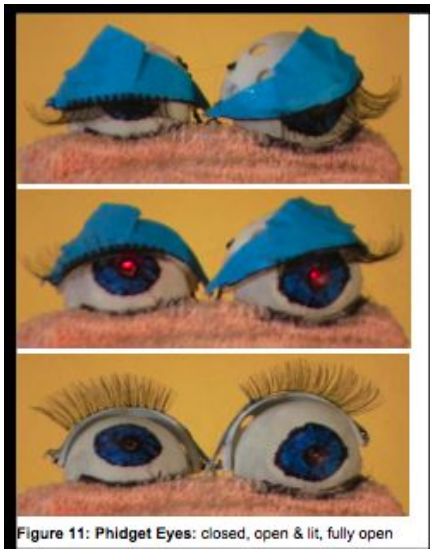
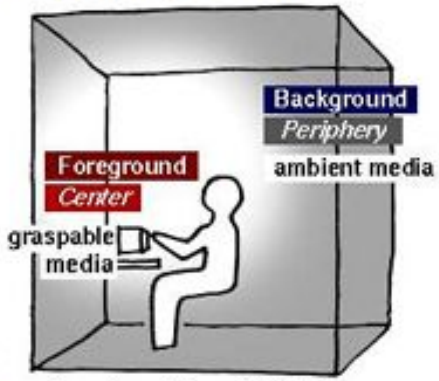


Figure 11: Phidget Eyes: closed, open & lit, fully open

Les interfaces tangibles

Tangible Bits is our vision of Human Computer Interaction (HCI) which guides our research in the Tangible Media Group. People have developed sophisticated skills for sensing and manipulating our physical environments. However, most of these skills are not employed by traditional GUI (Graphical User Interface). Tangible Bits seeks to build upon these skills by giving physical form to digital information, seamlessly coupling the dual worlds of bits and atoms.

Guided by the Tangible Bits vision, we are designing "tangible user interfaces" which employ physical objects, surfaces, and spaces as tangible embodiments of digital information. These include foreground interactions with graspable objects and augmented surfaces, exploiting the human senses of touch and kinesthesia. We are also exploring background information displays which use "ambient media" -- ambient light, sound, airflow, and water movement. Here, we seek to communicate digitally-mediated senses of activity and presence at the periphery of human awareness. The goal is to change the "painted bits" of GUIs (Graphical User Interfaces) to "tangible bits," taking advantage of the richness of multimodal human senses and skills developed through our lifetime of interaction with the physical world.

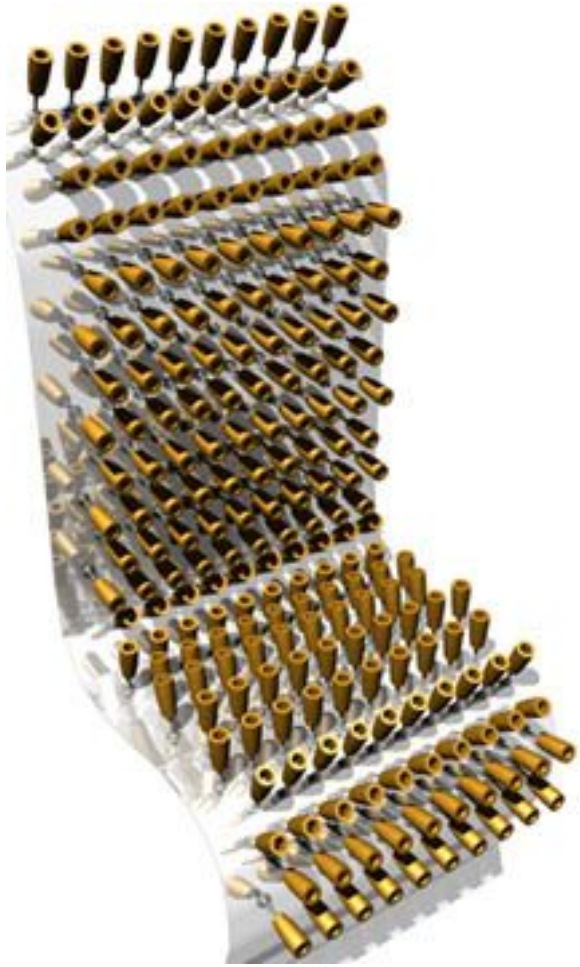


drawing: Hiroshi Ishii

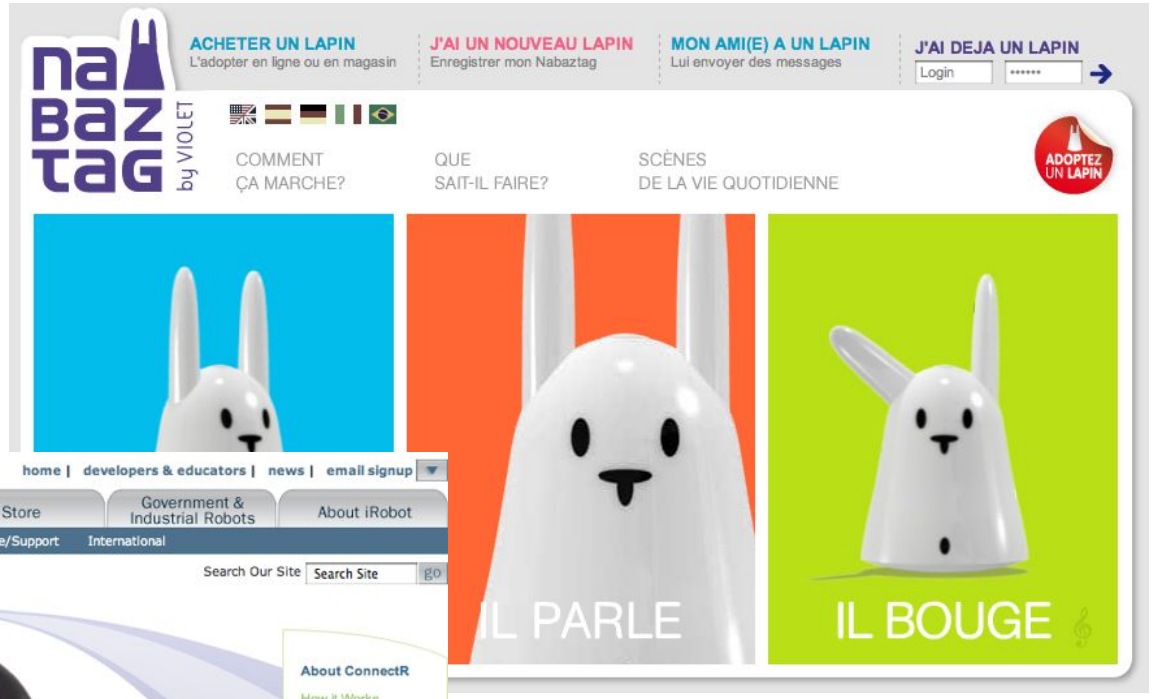
Tangible Bits full paper presented at CHI 97

Prof. Hiroshi ISHII http://web.media.mit.edu/~ishii/

Ex. de projet de l'équipe : super cilia skin

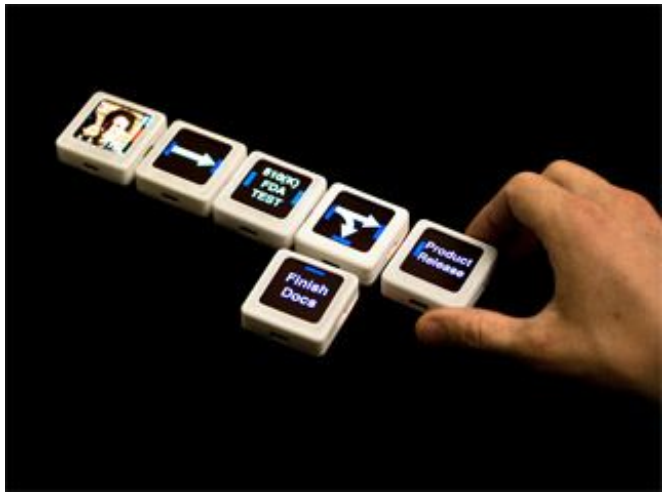


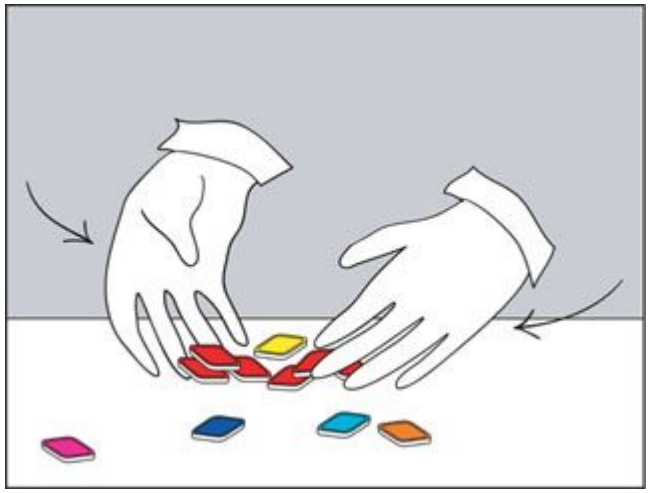
en 2005 : déjà une industrie !

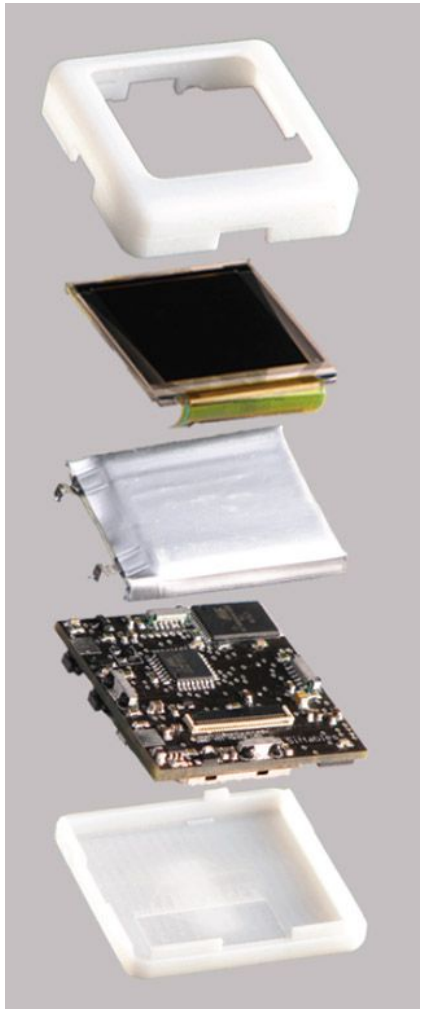


skyscout

Le projet « siftables » (David Merril, MIT, 2007)







http://web.media.mit.edu/~dmerrill/siftables.html

LOGIN CART

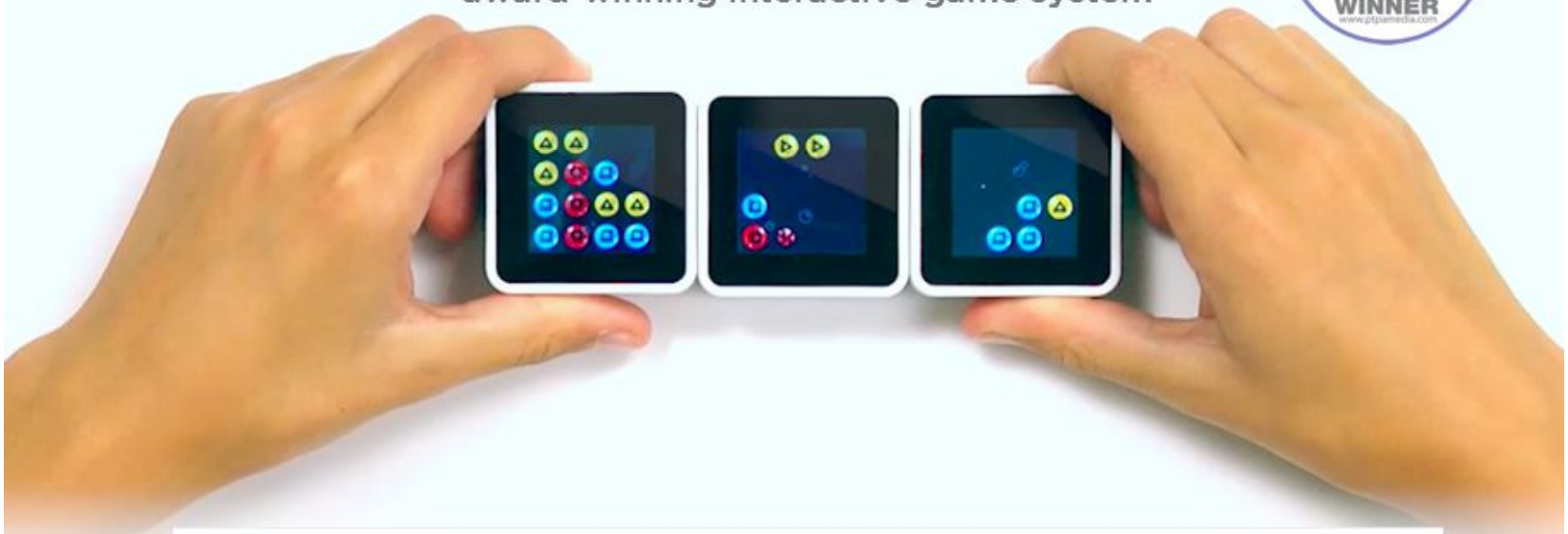
sifteo

Sifteo Cubes | Intelligent Play | Games | About Us | Press | Shop

Sifteo Cubes

award-winning interactive game system

parent tested parent approved WINNER



thought possible."

San Francisco Chronicle

"a clever new way for childre

projet "fat and furious" Master ENJMIN 2013

sifteo ♥ 3D Robotics

So long, thank you, we're still here

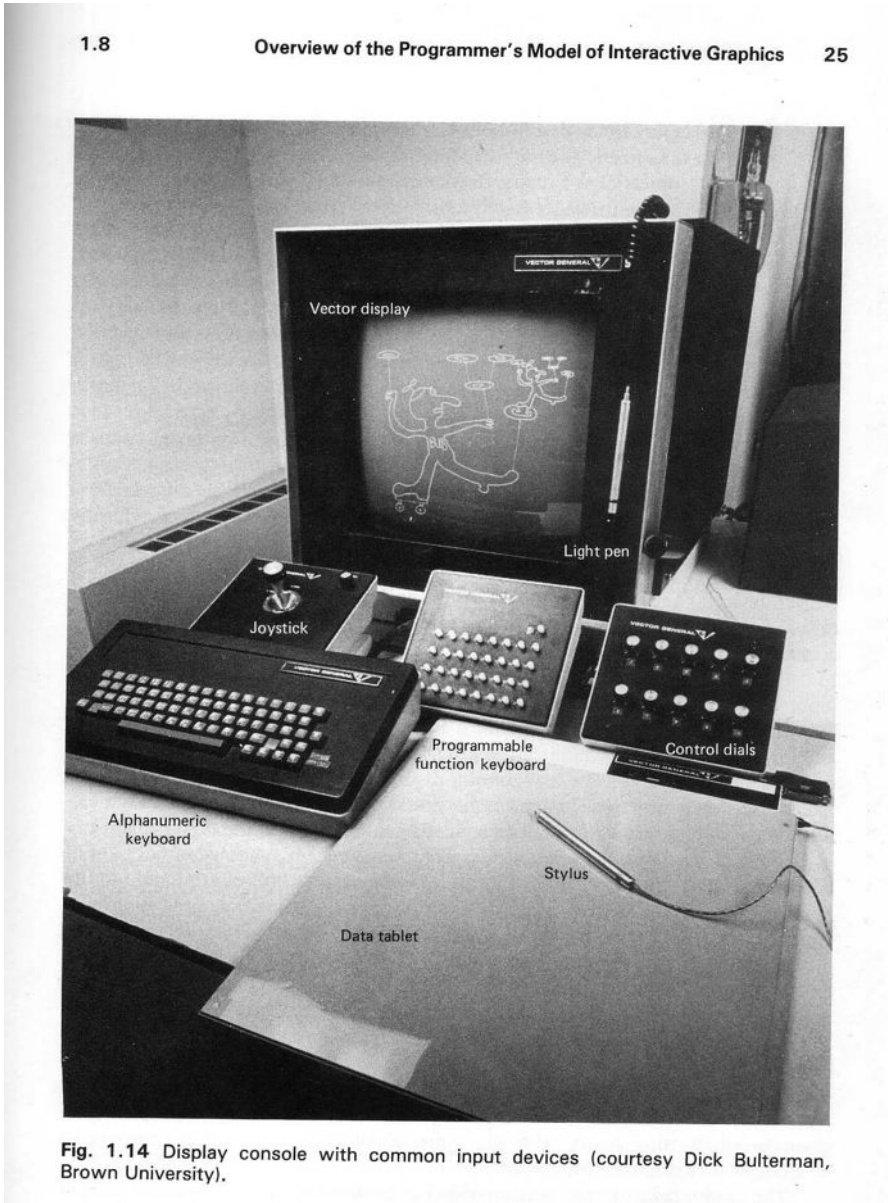
We'll cut to the chase: Sifteo has been acquired by 3D Robotics! We're really excited about it; we will continue to support Sifteo Cubes and Sifteo users; and we are so grateful to our customers and supporters around the world. *Thank you.*

handbuilt prototypes from our MIT days

When we (Dave and Jeevan) started Sifteo in 2009, we knew we could deliver magical versions of everyday objects - that we could use the latest in computing and sensors to create new interactions that were more natural, more human, and just better than what currently existed.

1) Périphériques "riches"

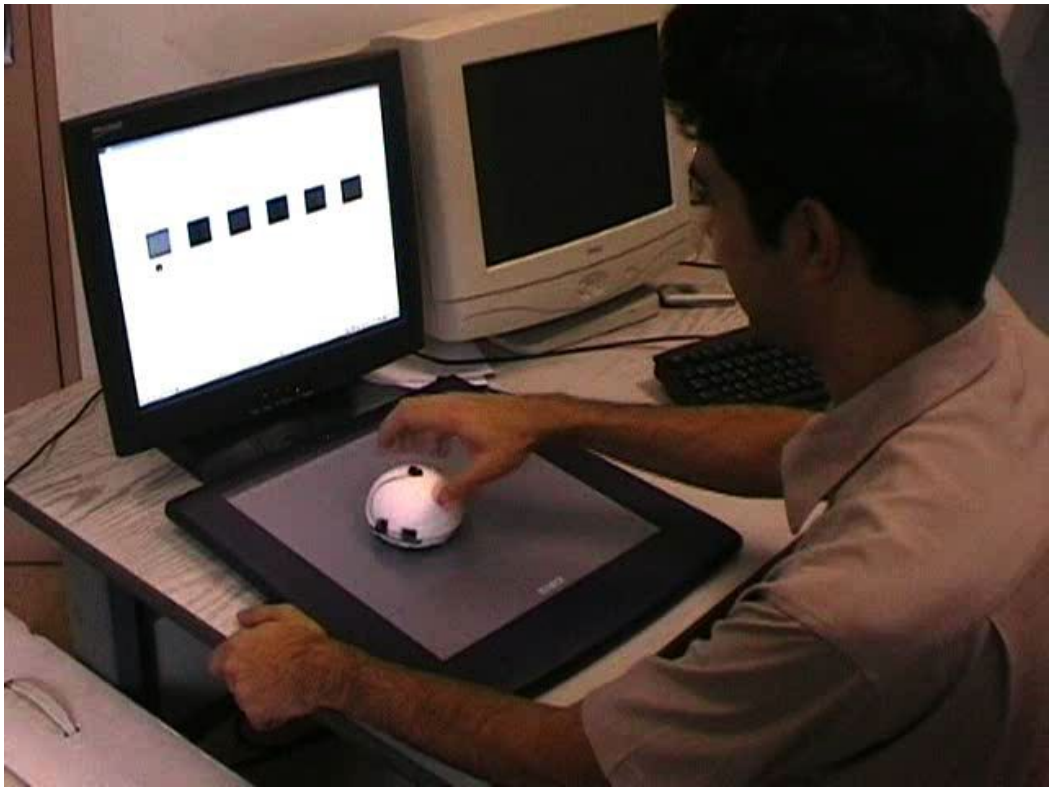
Rappel



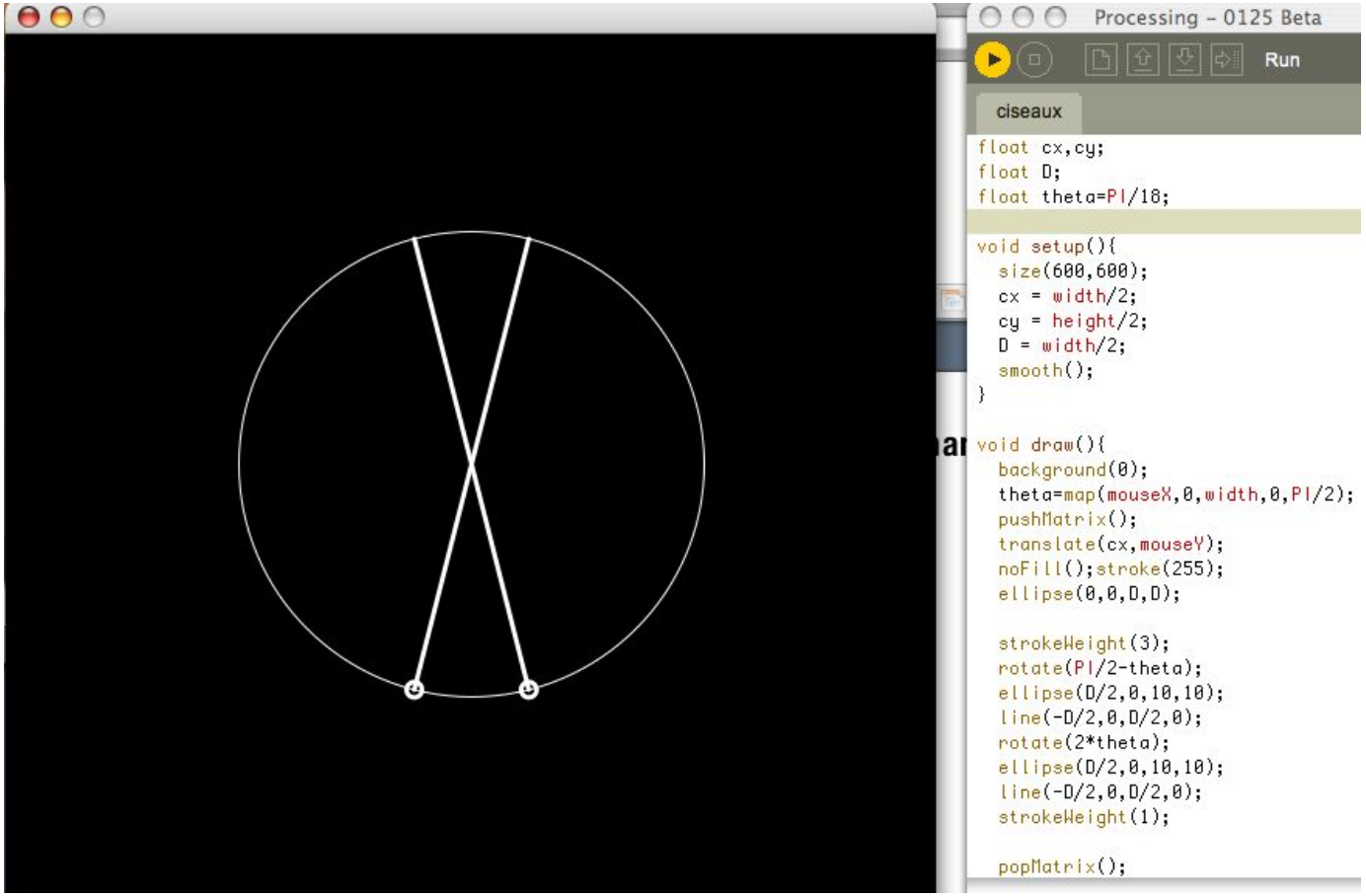
(Foley & van Dam, 1983)

La librairie procontrol

Ex : Souris à 3 DDL, Rodrigo Almeida, CNAM



Interaction bimanuelle : ex. des ciseaux

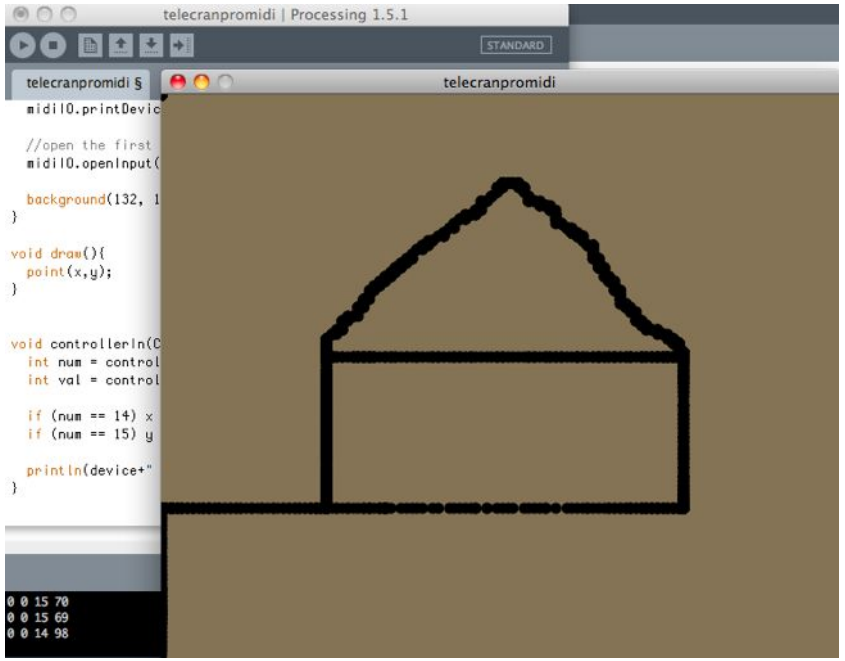


La librairie promidi

exemple en musique numérique : KORG nanokontrol

le télécran

```
import promidi.*;
MidiIO midiIO;

int x, y = 0;

void setup(){
  size(600,500);
  smooth();
  strokeWeight(10);

  //get an instance of MidiIO
  midiIO = MidiIO.getInstance(this);
  println("printPorts of midiIO");

  //print a list of all available devices
  midiIO.printDevices();

  //open the first midi channel of the first device
  midiIO.openInput(0,0);

  background(127);
}

void draw(){
  point(x,y);
}

void controllerIn(Controller controller, int device, int channel){
  int num = controller.getNumber();
  int val = controller.getValue();

  if (num == 14) x = int(map(val,0,127,0,width));
  if (num == 15) y = int(map(val,127,0,0,height));

  println(device+" "+channel+" "+num+" "+val);
}
```

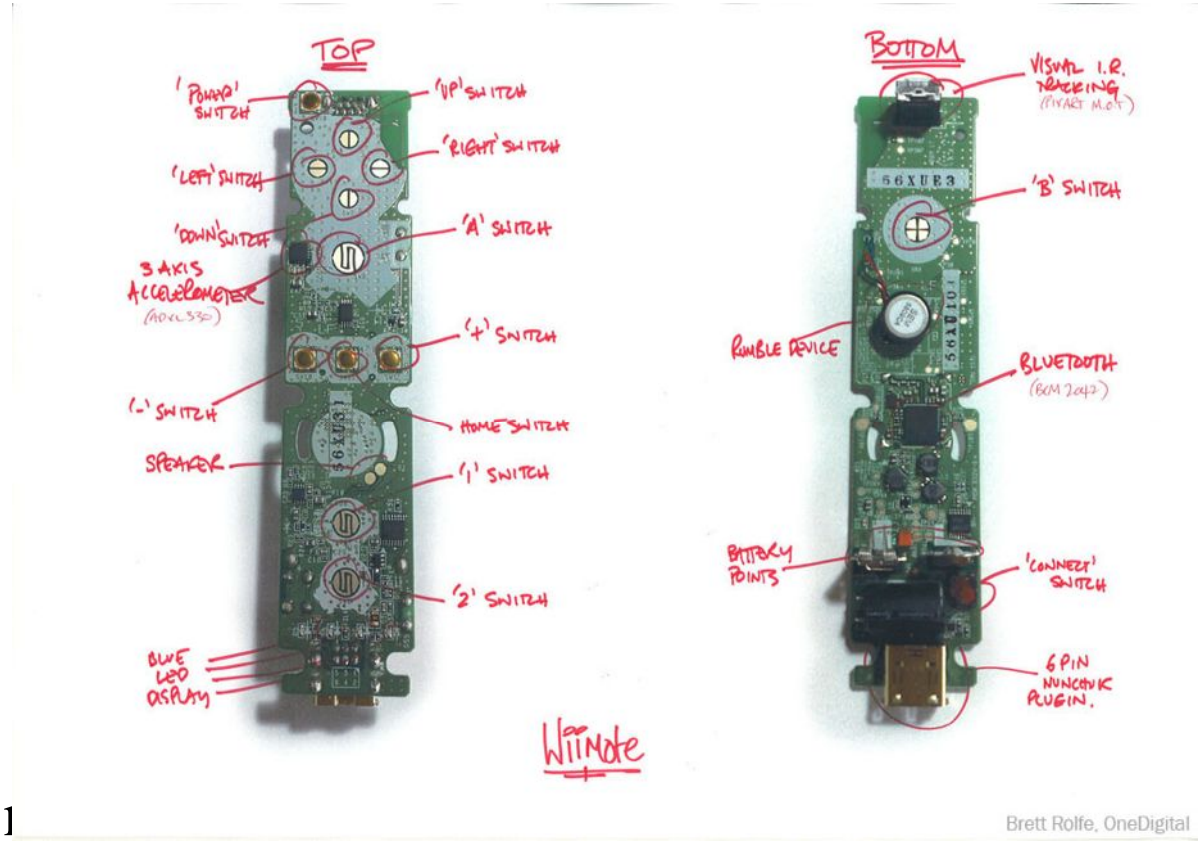


gestion de la cohérence du mapping ?

Mon ampli cambridge audio A650 : bouton motorisé

2) la wiimote

La Wiimote de Nintendo (2006)



40 € wiimote + 20 €

-accelerometre 3axes

-Camera IR + rec. Blobs

-HP, vibreur

-Plein de boutons + joysticks

-Bluetooth (et i2c avec le nunchuck)

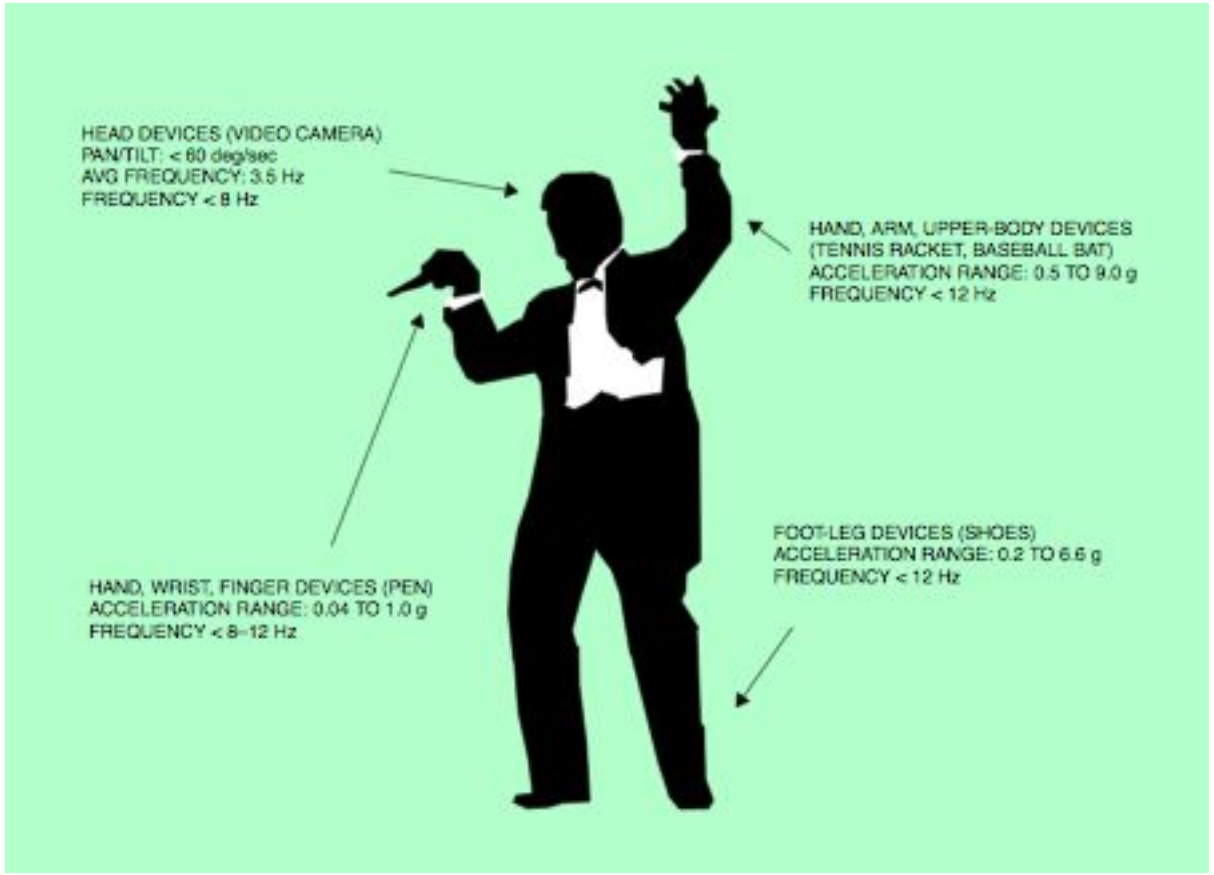
Totalement « hacké »

=> www.wiili.com

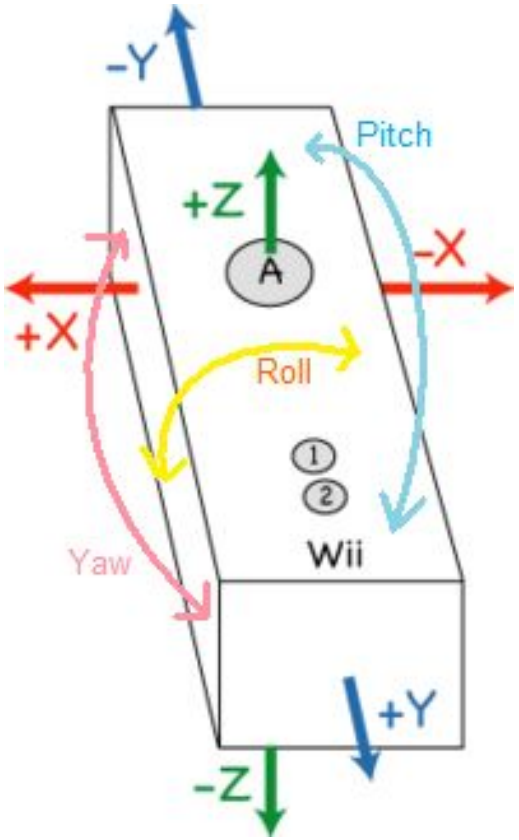
C. Verplaetse IBM Systems Journal 35(3-4) **1996** !!

Inertial proprioceptive devices: Self-motion-sensing toys and tools

by C. Verplaetse



Utilisation de l'accéléromètre



On pose la wiimote de manière à avoir successivement les trois axes X Y Z à la verticale et on collecte les valeurs renvoyées.

+Z : x1, y1, z1
+Y : x2, y2, z2
+X : x3, y3, z3

d'où les coordonnées du point origine:

x0 = (x1 +x2)/2
y0 = (y1 + y3)/2
z0 = (z2 + z3) /2

On obtient alors les coordonnées du vecteur force (exprimées en g) :

ax = (xraw -x0)/(x3-x0)
ay = (yraw - y0)/(y2-y0)
az = (zraw - z0)/(z1-z0)

http://www.wiili.org/index.php/Motion_analysis

Pour le nunchuk, il est plus difficile de faire cette calibration du fait du facteur forme. L'examen de photos d'amateurs sur le Web permet d'éviter un démontage et suggère que l'accéléromètre est placé sur un plan horizontal lorsque le nunchuck est pris en main comme un pistolet.

Calcul d'orientation pitch et roll :

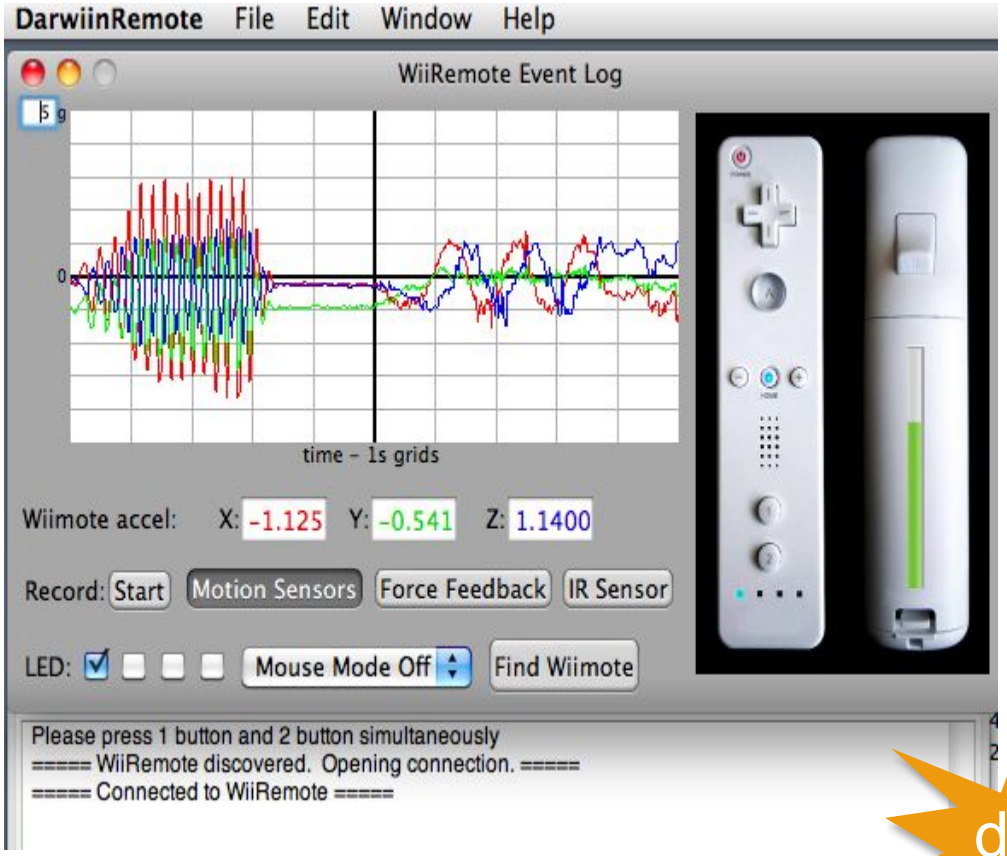
Si la wiimote n'est pas en mouvement accéléré, la mesure des cordonnées du vecteur gravité permet de d'obtenir l'orientation dans l'espace du dispositif en pitch et roll :

pitch = arctan(ax / sqrt(ay*ay + az*az))
roll = arctan(ay / sqrt(ax*ax + az*az))

source : Kionix. *Tilt-sensing with Kionix MEMS Accelerometers*. Application note AN005. 2005 (en ligne sur www.kionix.com)

On peut vérifier le repos en s'assurant que la norme ax*ax + ay*ay + az*az est proche de l'unité.

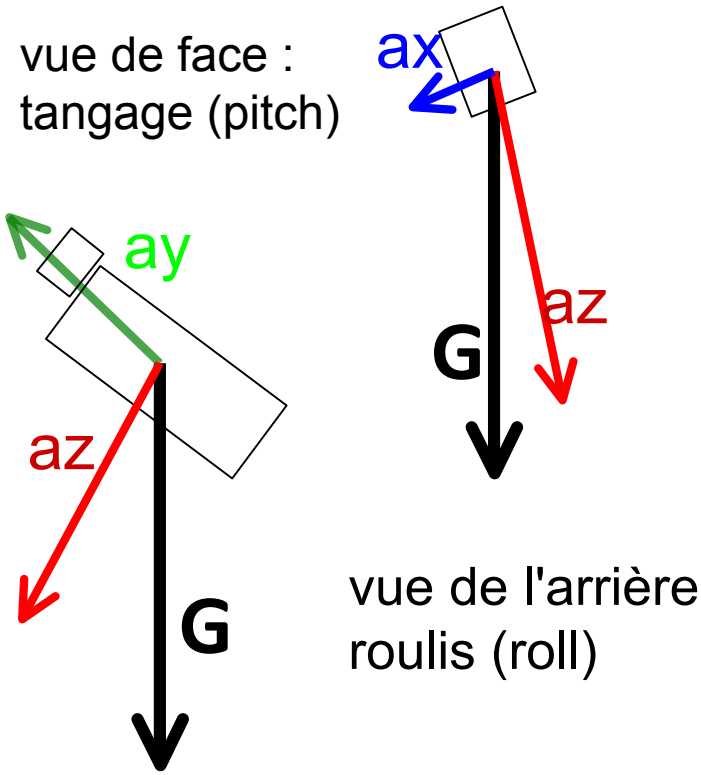
Utilisation de l'accéléromètre



démo

application à la mesure d'angles

ma (belle) lunette astronomique



$$tg(pitch) = ay/az$$

$$tg(roll) = ax/az$$

demowii

```
import lll.wrj4P5.*;
import lll.Loc.*;
import wiiremotej.*;
import wiiremotej.event.*;
Wrj4P5 mawii;

void setup(){
 size(600,600);fill(255);noStroke();smooth();
 rectMode(CENTER);
 mawii=new Wrj4P5(this).connect();
}

void draw(){
 if (mawii.isConnecting()) return;
 Loc s = mawii.rimokon.sensed;
 float pitch = atan2(s.z,s.y);
 float roll = atan2(s.z, s.x);
 background(100);
 pushMatrix();
 translate(300,300);
 rotate(pitch);
 rect(0,0,50,300);
 popMatrix();
}
```

Phase préalable de calibration

```
 import lll.wrj4P5.*;
import lll.Loc.*;
import wiiremotej.*;
import wiiremotej.event.*;

/// variables pour wrjp5
Wrj4P5 mawii;

/// variables pour moi
float ax, ay, az;
float K;
int N;

 void setup() {
  N = 0;
  K = 1.0;
  ax = ay = az = 0.0;
   mawii=new Wrj4P5(this).connect();
   frameRate(20);
}
```

```
void draw() {
 if (mawii.isConnecting()) return;
 Loc s = mawii.rimokon.sensed;
 //println(s.x+" "+s.y+" "+s.z);
 ax = (1-K)*ax + K*s.x;
 ay = (1-K)*ay + K*s.y;
 az = (1-K)*az + K*s.z;
 K = K/(K+1);
 if (N % 400 == 0) println(N+"   "+ax+"       "+ay+"       "+az);
 N++;
}

void keyPressed(){
 println("\n !!! on rejoue !!!\n");
 N = 0;
 K = 1;
 ax = ay = az = 0;
}
```

Calcul des valeurs moyennes avec le capteur stable sur 6 positions de référence pour (ax, ay, az) : (0,1,0) (0,-1,0) (0,0,1) (0,0,-1) (1,0,0) (-1,0,0)

```
#### debout

0   0.00952381      -1.0096154      0.02970297
400   0.0094050625      -0.99832153      -0.008617069
800   0.010070753      -0.99590683      -0.008417703
1200   0.010189927      -0.9937544      -0.008614934

 !!! on rejoue !!!

### a plat

0   -0.00952381      -0.009615385      1.019802
400   -0.004797531      0.0033090361      0.9940255
800   -0.0041257925      0.004177467      0.9919165
1200   -0.004488322      0.0048597306      0.9904531
1600   -0.0048303152      0.005014891      0.9900685

 !!! on rejoue !!!

### de coté

0   -0.96190476      -0.009615385      0.02970297
400   -0.9596727      -0.011221944      0.028715342
800   -0.9601448      -0.011452028      0.028862452
1200   -0.96011156      -0.01177704      0.028878579
1600   -0.9604343      -0.01221592      0.028911388
```

Calcul des angles avec filtrage des données

Boucle filtrage alpha très fort :

estimé = (1-alpha)*estimé + alpha*mesure

alpha fixé
a priori

Boucle de filtrage de Kalman :

K = (P+Q)/(P+Q+R)
estimé = (1-K)*estimé*+K*mesure
P = (1-K)*(P+Q)

R = ecart type bruit
Q = ecart type processus
Pinitial = 1

choix à faire entre réponse et stabilité ….

Le programme complet

```
import lll.wrj4P5.*;
import lll.Loc.*;
import wiiremotej.*;
import wiiremotej.event.*;

/// variables pour wrjp5
Wrj4P5 mawii;

/// variables pour moi
PFont font;
float ax,ay,az,pitch,roll;
PFont metaBold;
float P = 1.0;
float K;
float R = 0.01;
float Q = 0.001; //0.0001 pour un filtrage plus fort
float rolle = 0.0;
float pitche = 0.0;
float rolla = 0.0;
float pitcha = 0.0;

void setup() {
  size(800,300);
  mawii=new Wrj4P5(this).connect();
  font = loadFont("Helvetica-32.vlw");
  textFont(font);
  //textMode(SCREEN);
  frameRate(20);
  smooth();
}

void draw() {
  if (mawii.isConnecting()) return;
  // acquisiton et recalibration
   Loc s = mawii.rimokon.sensed;

   ax = map(s.x, -0.9624, 0.9877, -1.0, 1.0);
   ay = map(s.y, -1.0045, 1.0067, -1.0, 1.0);
   az = map(s.z, -0.9987, 0.9886, -1.0, 1.0);

    // angles
   pitch = atan2(sqrt(ay*ay+az*az),ax);
   roll = atan2(sqrt(ax*ax+az*az),ay);
   // conversion en degres
   pitch = map(pitch, 2*PI/180, 178*PI/180, 0, PI);
   roll = map(roll, 2*PI/180, 178*PI/180, 0, PI);

   // filtrage Kalman
   K = (P+Q)/(P+Q+R);
   rolle = rolle*(1.0-K)+K*roll;
   pitche = pitche*(1.0-K)+K*pitch;
   P = (1.0-K)*(P+Q);

   // ou bien filtrage alpha tres fort

   rolla = 0.02*roll + 0.98*rolla;
   pitcha = 0.02*pitch + 0.98*pitcha;

 // affiche les infos
 background(255);
 fill(0);
 text("ROLL  : "+str(roll*180/PI), 50, 50);
 fill(0,0,225);
 text("ROLLa : "+str(rolla*180/PI), 50, 100);
 fill(0);
 text("PITCH : "+str(pitch*180/PI), 400, 50);
 fill(0,0,225);
 text("PITCHa: "+str(pitcha*180/PI), 400, 100);

// dessin

 translate(200,200);
 stroke(175);noFill();
 ellipse(0,0,160,160);
 stroke(255,0,0);
 rotate(PI/2-roll);
 line(-50,0,50,0);
 stroke(125);
 rotate(-PI/2+roll);
 rotate(PI/2-rolle);
 strokeWeight(3);line(-80,0,80,0);strokeWeight(1);
 stroke(0,0,225);
 rotate(-PI/2+rolle);
 rotate(PI/2-rolla);
 strokeWeight(3);line(-80,0,80,0);strokeWeight(1);
 resetMatrix();

///// idem pour les autres angles pitch, pitche, pitcha
 translate(600,200);
 stroke(175);noFill();
 ellipse(0,0,160,160);
 stroke(255,0,0);
 rotate(PI/2-pitch);
 line(-50,0,50,0);
 stroke(125);
 rotate(-PI/2+pitch);
 rotate(PI/2-pitche);
 strokeWeight(3);line(-80,0,80,0);strokeWeight(1);
 stroke(0,0,225);
 rotate(-PI/2+pitche);
 rotate(PI/2-pitcha);
 strokeWeight(3);line(-80,0,80,0);strokeWeight(1);
 resetMatrix();
 }
```

Xsens

Autres produits du même genre:

Sparkfun 450$

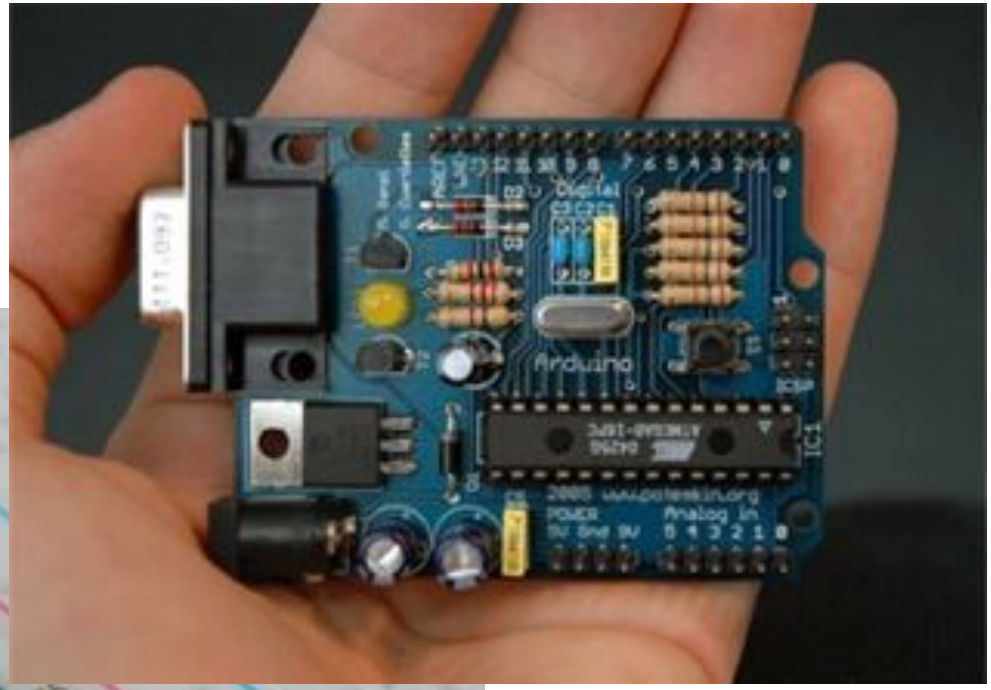
http://www.sparkfun.com/commerce/product_info.php?products_id=8454

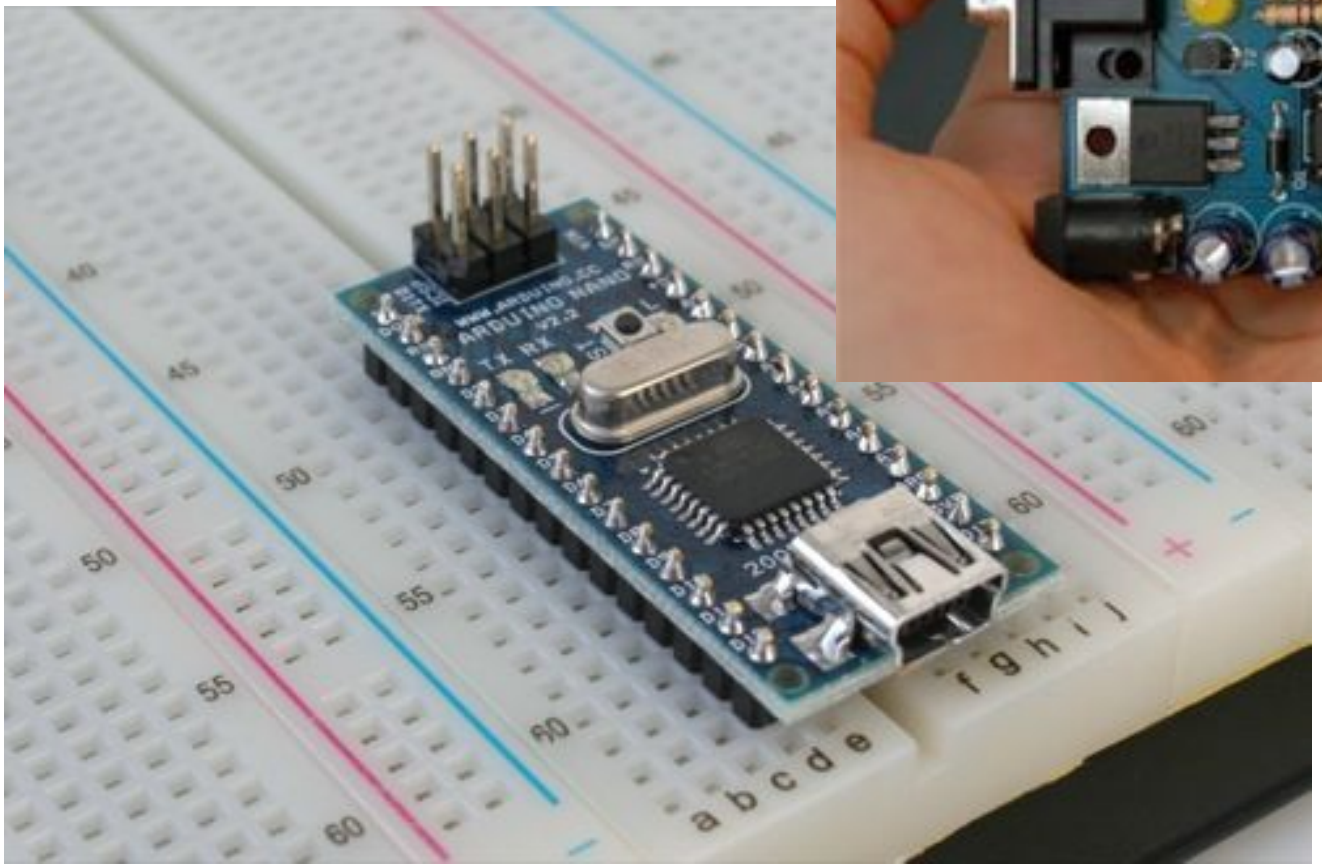
à voir : les expériences de J.C. Lee (HCII, Carnegie Mellon Univ.)

3. La carte ARDUINO

(Science et vie junior - février 2012)

www.arduino.cc





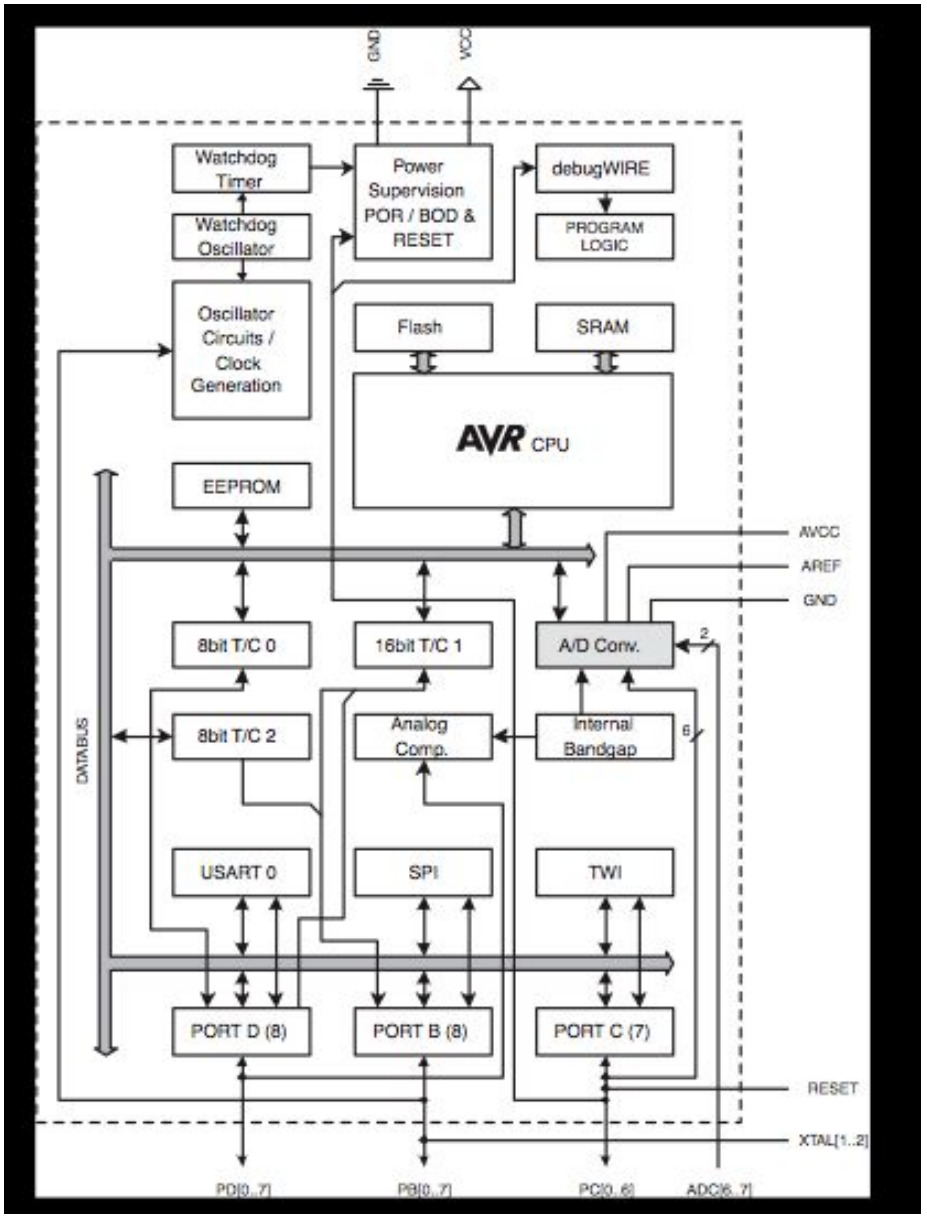
+ une version bluetooth

En France : AlyaSoft (à Bézier) : 22 euros + lextronic + st quentin radio

Le μcontroleurs ATMega
de ATMEL:

Risc 24 MIPS / 24 MHz horloge
131 instructions, 32*8 registres
16 Ko FLASH + 1 Ko SRAM
+ 512b EEPROM

La carte ARDUINO reprend
toutes ses E/S :
-6 E. analogiques 10bits
-6 E/S numériques (PWM)
-port série
+ alim
+ port usb vers l'hôte



L'environnement Arduino

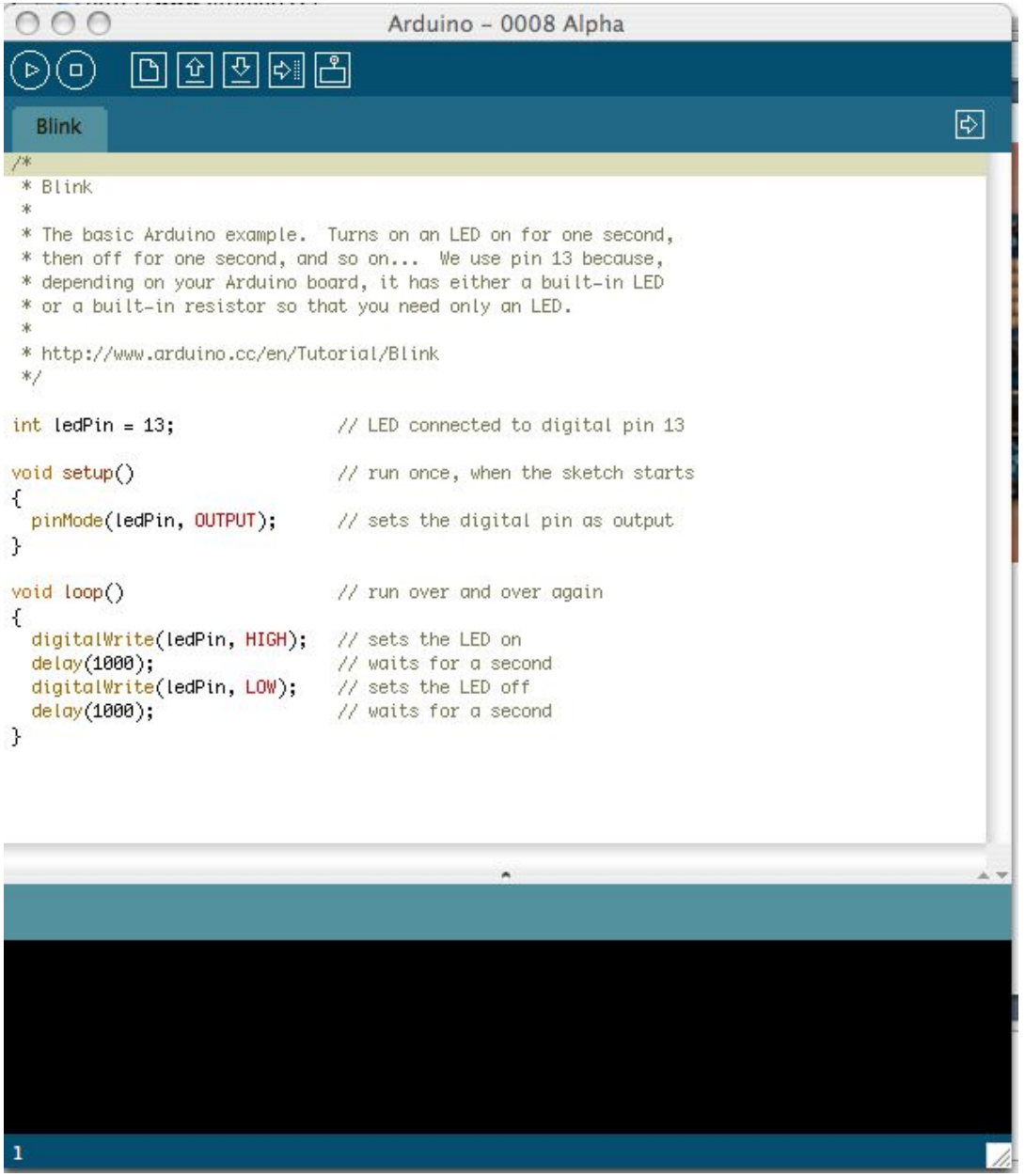
Multiplateforme !
Open source !

Basé sur C/C++

Types flottants !!
fcts math
tableaux

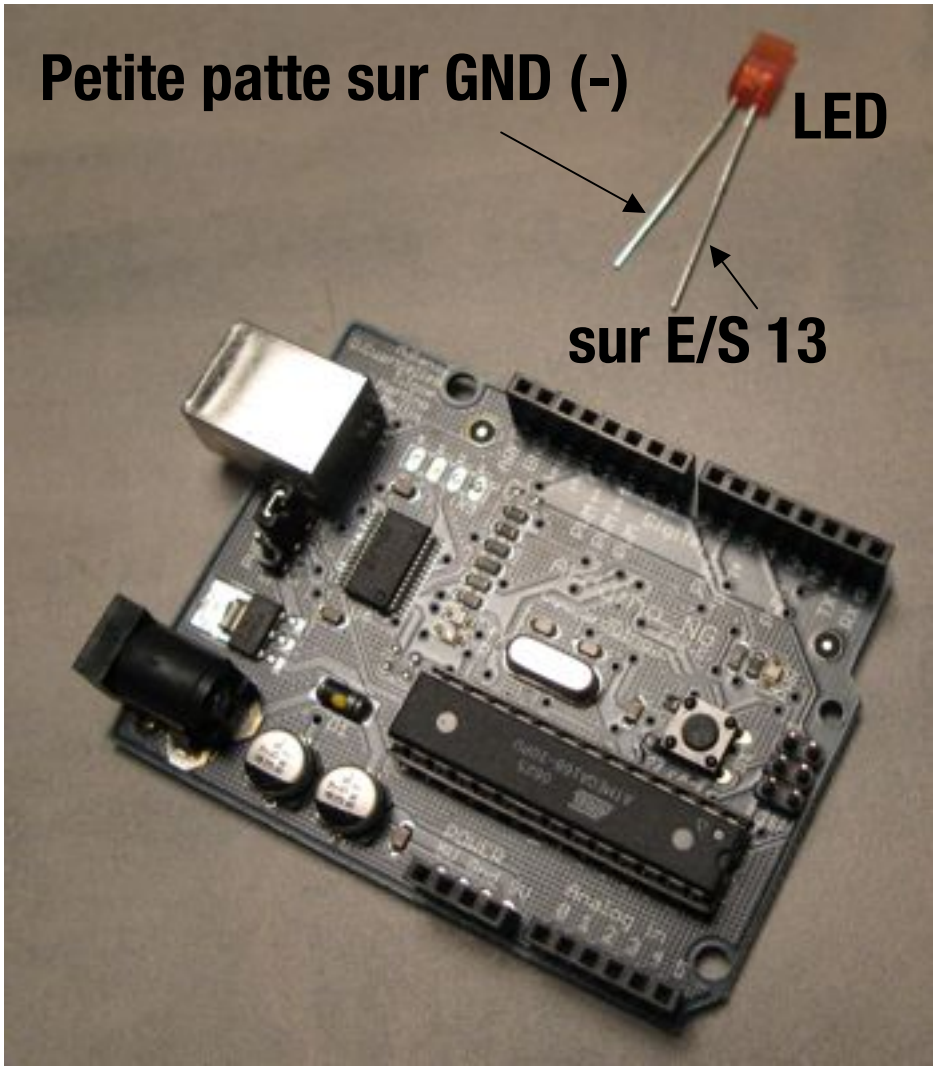
Librairies de com
Serie, i2c (Wire)

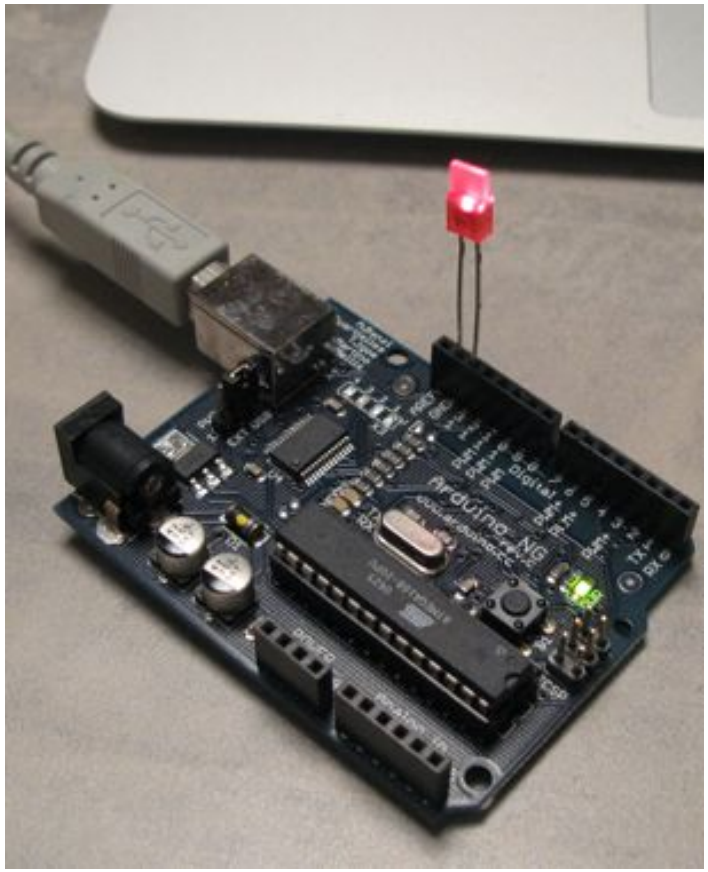
Contrôle servos …



« blink » : le Hello world de l'Arduino

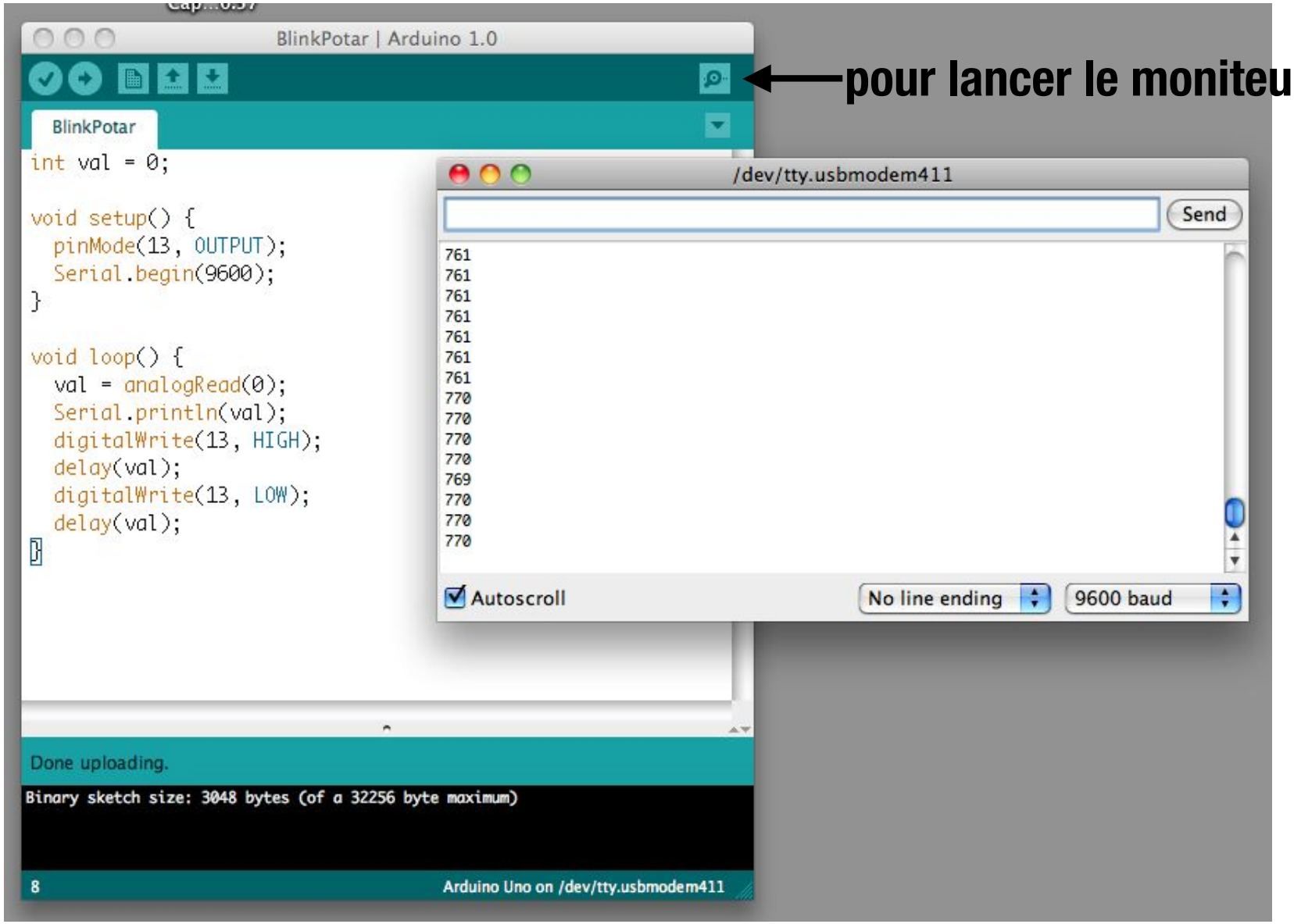
Blink : le montage



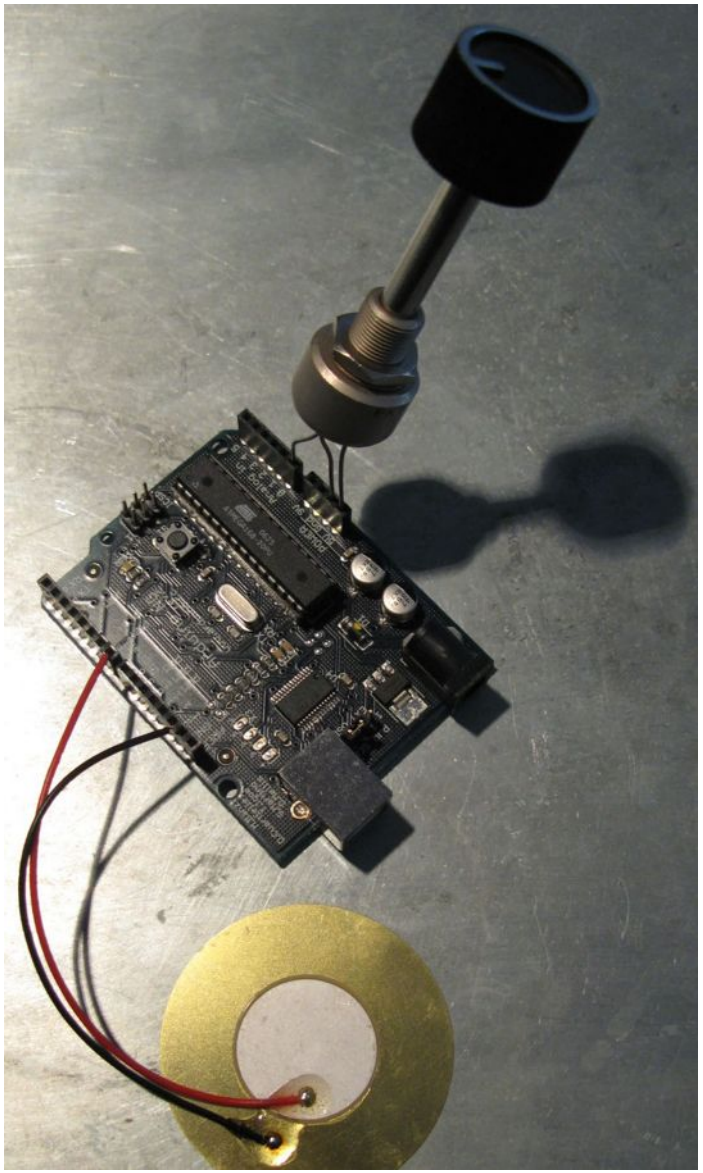


Fiat lux …

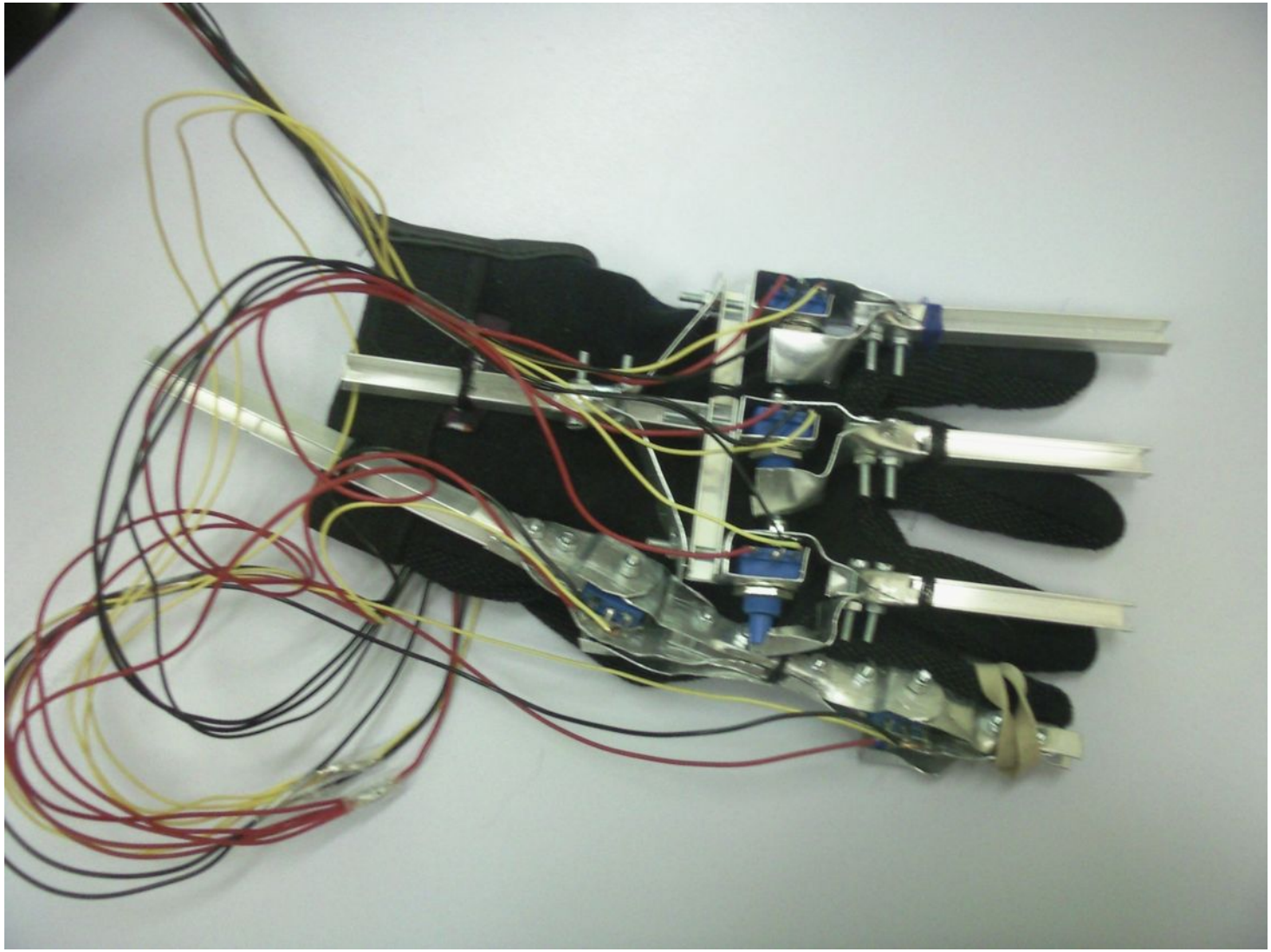
Entrée analogique (et liaison série)



Buzzer :

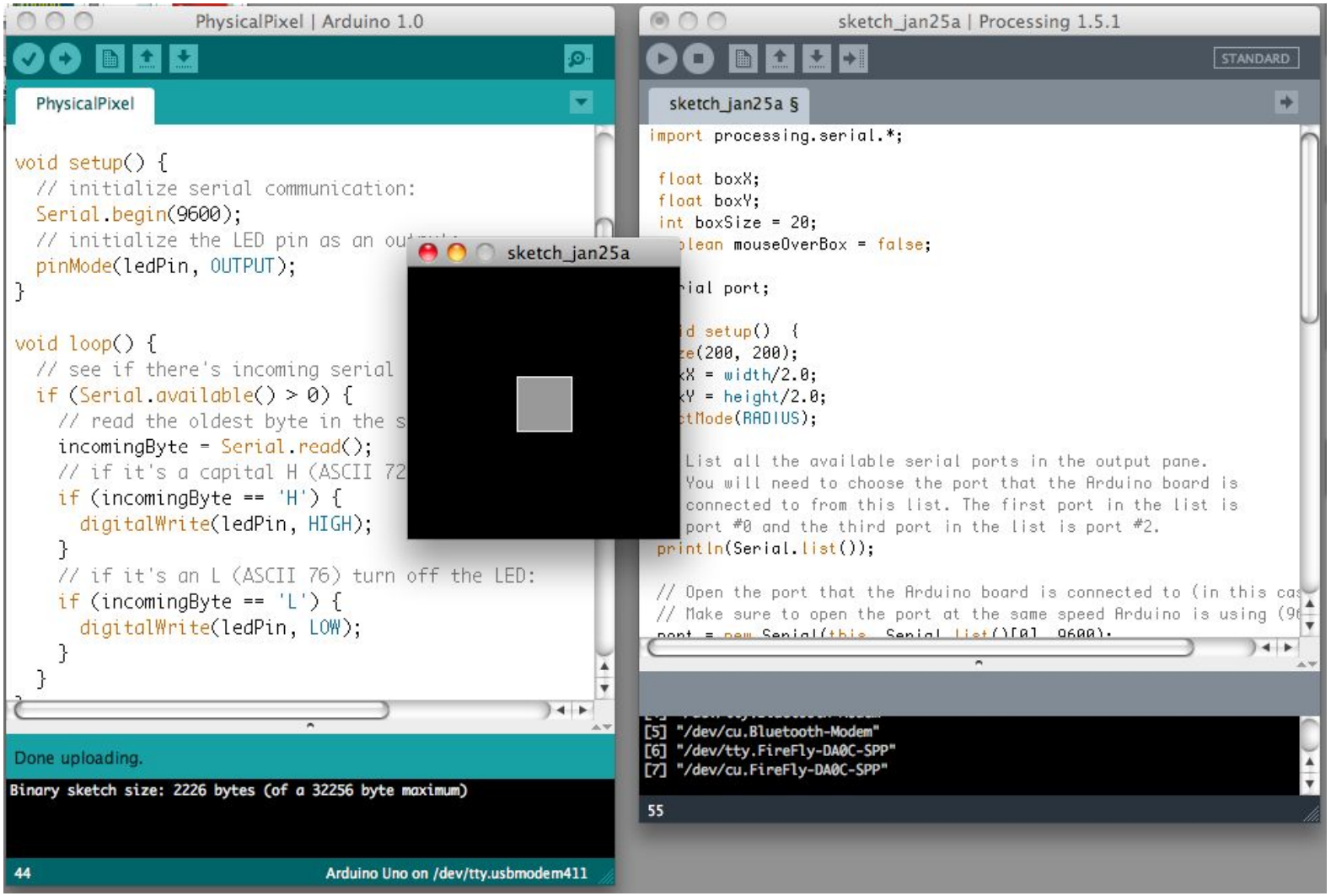


```
/* copyleft 2006 Tod E. Kurt <tod@todbot.com
 * http://todbot.com/
 */

int potPin = 0;    // select the input pin for the potentiometer
int speakerPin = 7;

int val = 0;

void setup() {
  pinMode(speakerPin, OUTPUT);
}

void loop() {
  digitalWrite(speakerPin, LOW);
  val = analogRead(potPin);    // read value from the sensor
  val = val*2;                 // process the value a little
  for( int i=0; i<500; i++ ) {  // play it for 50 cycles
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(val);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(val);
  }
}
```

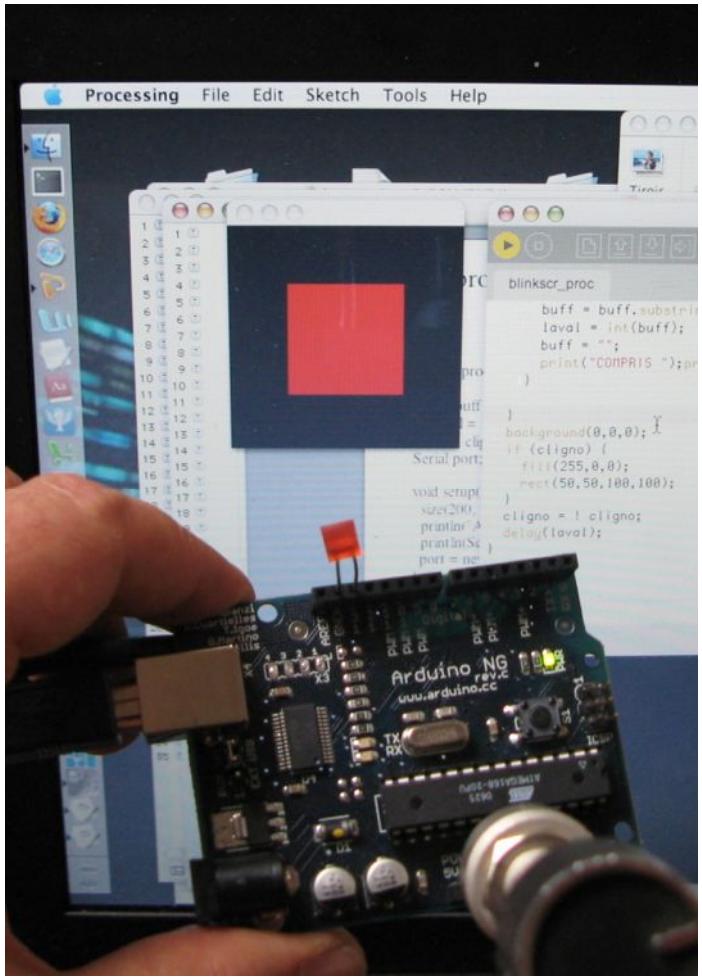


gant de captation (élève ingénieur cnam paris, 2010)

Dialogue avec une application processing (liaison série) : La demo "physical pixel" de exemples->communication



Blink, cette fois sur l'écran (avec processing)



Sur l'arduino =>

```
int potarPin = 0;
boolean ledState = LOW;
int laval;

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop()
{
  laval = analogRead(potarPin);
  Serial.println(laval);
  ledState = ! ledState;
  digitalWrite(13, ledState);
  delay(laval);
}
```

sur processing :

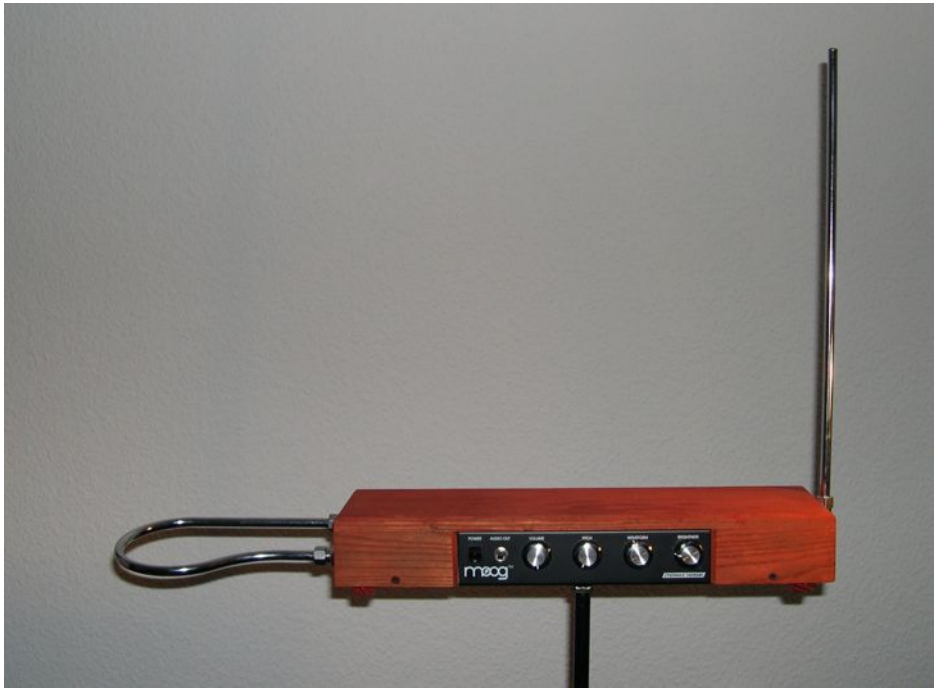
```
import processing.serial.*;

String buff = "";
int laval = 100;
boolean cligno = true;
Serial port;

void setup() {
  size(200, 200);
  println("Available serial ports:");
  println(Serial.list());
  port = new Serial(this, "/dev/tty.usbserial-A50018tg", 9600);
}
```

```
void draw() {
  if (port.available() > 0) {
   String buff = port.readStringUntil(13);
   if (buff != null) {
      print("RECU ");println(buff);
      buff = buff.substring(0, buff.length()-1);
      laval = int(buff);
      buff = "";
      print("COMPRIS ");println(laval);
    }

  }
  background(0,0,0);
  if (cligno) {
    fill(255,0,0);
    rect(50,50,100,100);
  }
  cligno = ! cligno;
  delay(laval);
}
```

Rque : le println d'Arduino envoie un CR et LF après laval

Les deux processus synchrones ???

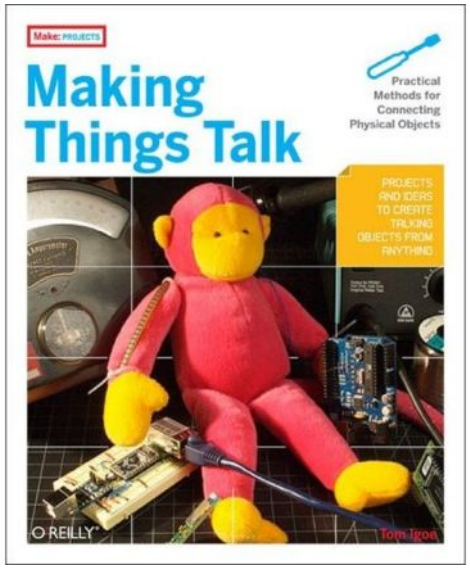
Un bon projet : (re)faire un "Theremin"

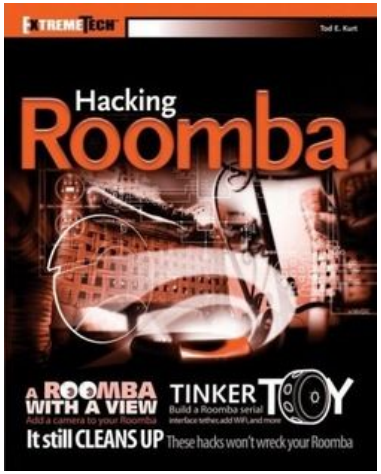


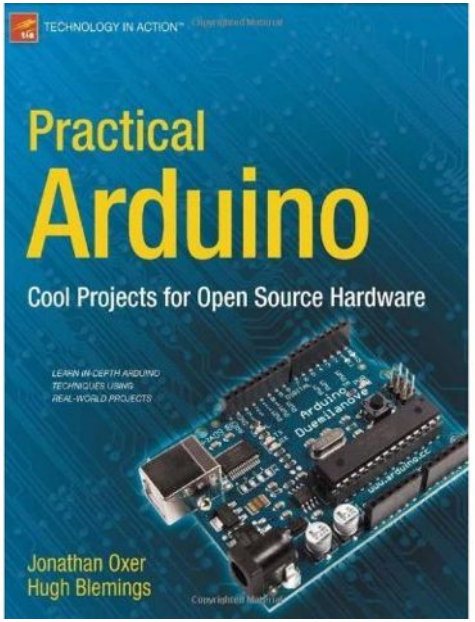
Pour continuer :

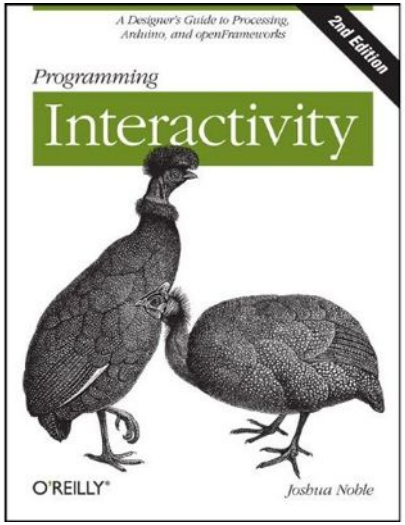
http://todbot.com/blog/spookyarduino/

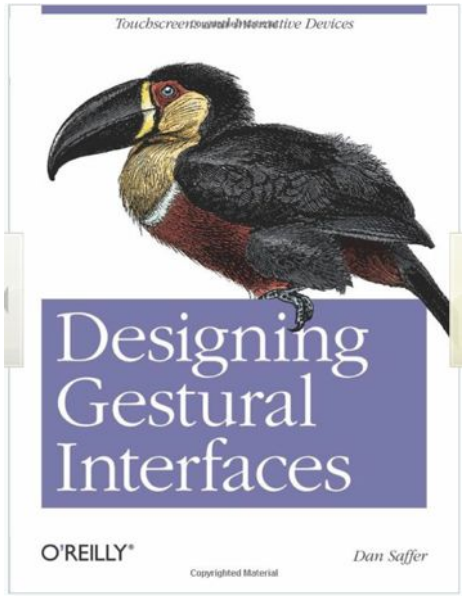
Bibliographie

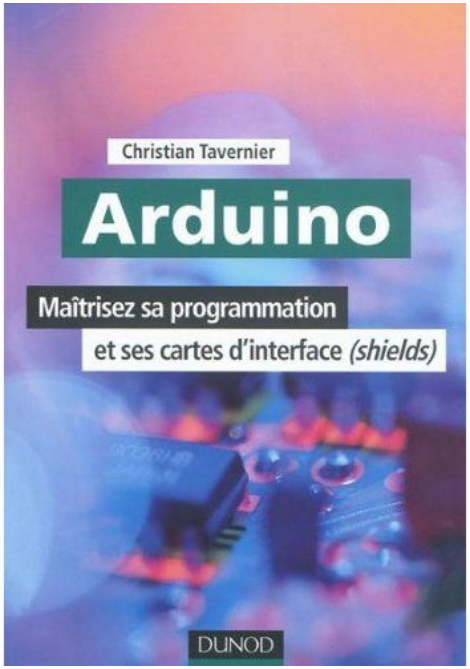




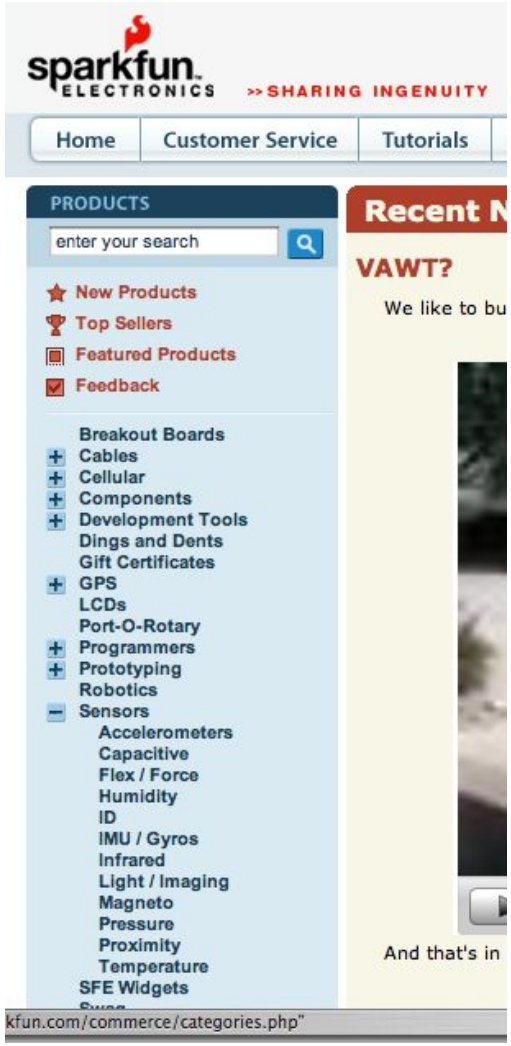








DIY (do it yourself) : les revendeurs



+ à Paris : StQuentin Radio, ACER, etc.

et à Shenzhen

Et un site (parmi plein)