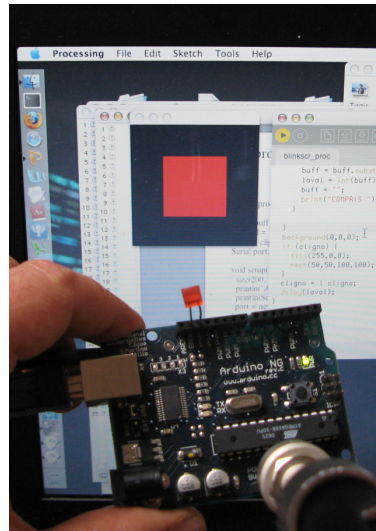


CNAM - NSY116 - janvier 2012

Un peu de physical computing

Pierre Cubaud
cubaud @ cnam.fr

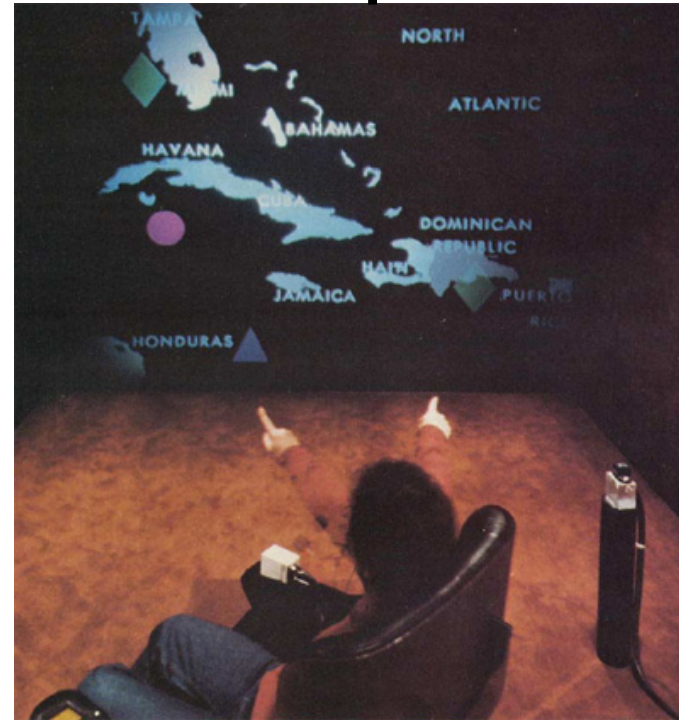
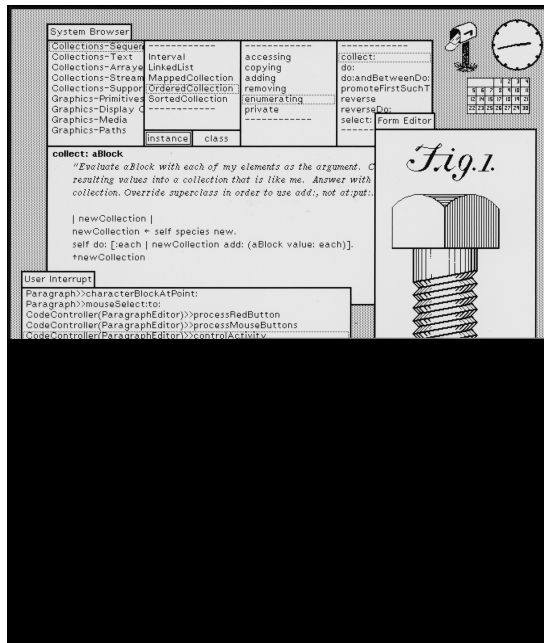


- Généralités
- La wiimote
- La carte Arduino
- Couplage
Arduino/processing

Bolt - put that there



1961 : Ivan Sutherland sur TX1 (MIT)



Wellner

L'IHM avant le WIMP

(Foley & van Dam, 1983)

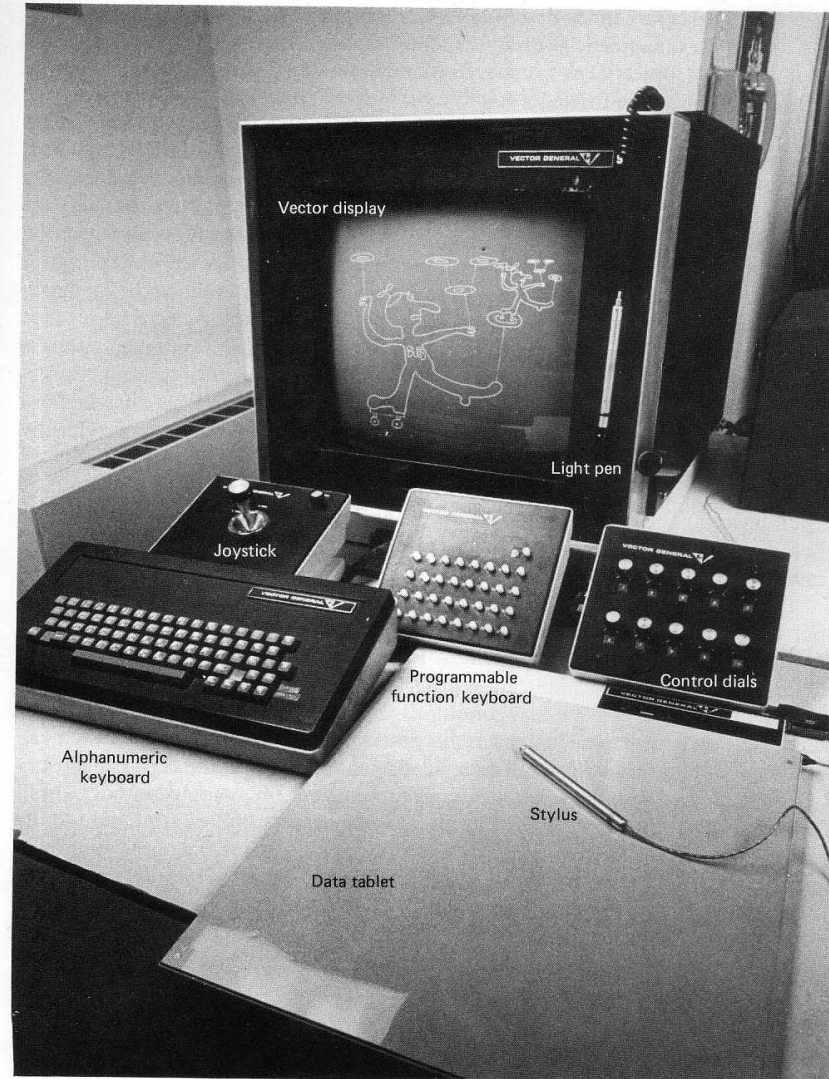


Fig. 1.14 Display console with common input devices (courtesy Dick Bulterman, Brown University).

Et aujourd'hui ??



exemple en musique numérique : KORG nanokontrol

Utilisation avec Processing (bibliothèque promidi)

```
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);

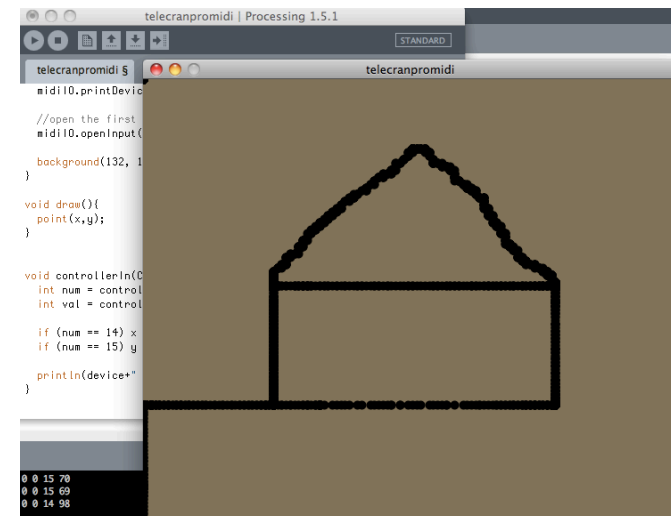
  background(127);
}

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

"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.

INTRODUCTION

(Greenberg, UIST'01)

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)

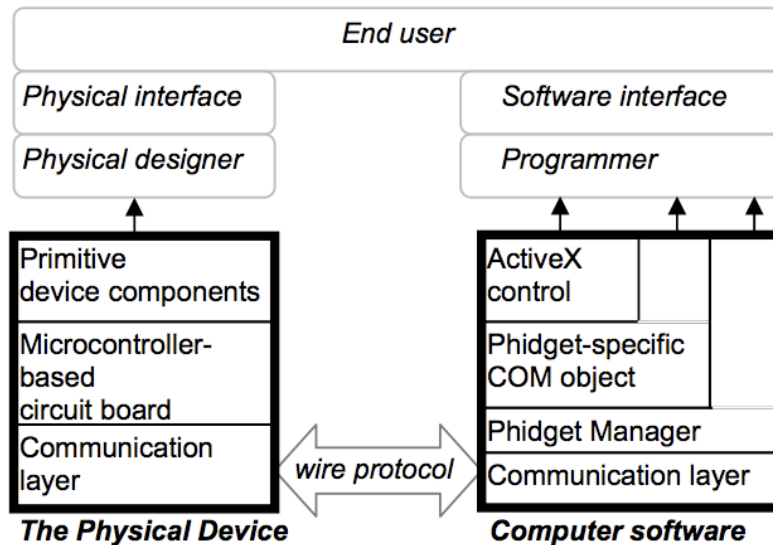


Figure 5. Phidget Architecture

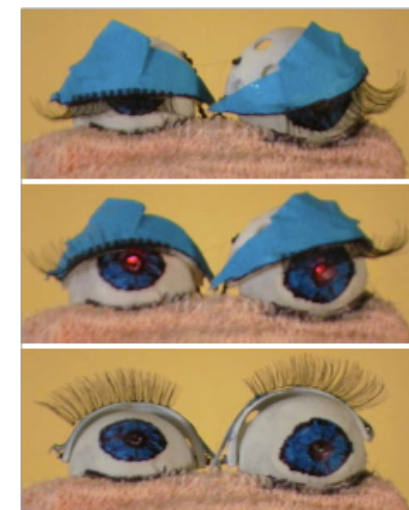
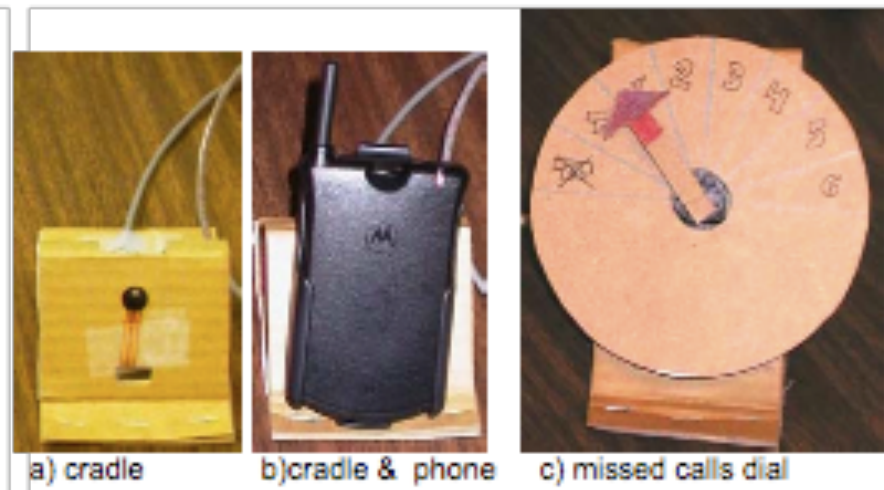
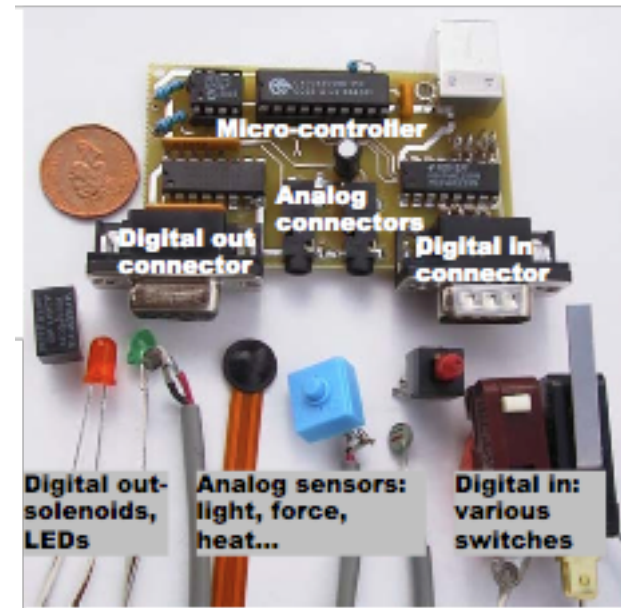
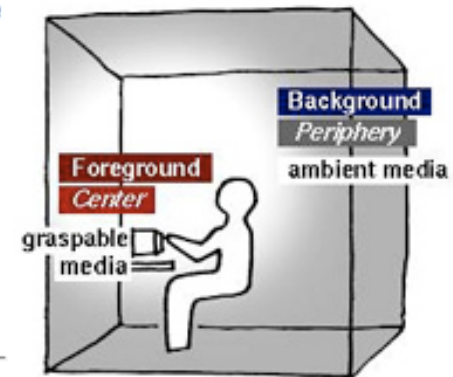


Figure 11: Phidget Eyes: closed, open & lit, fully open

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.

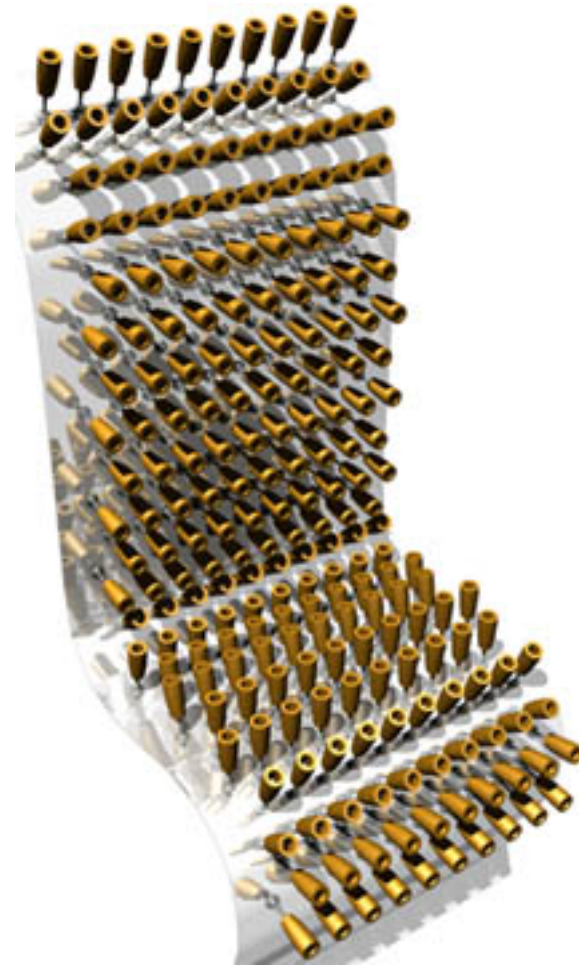


drawing: Hiroshi Ishii

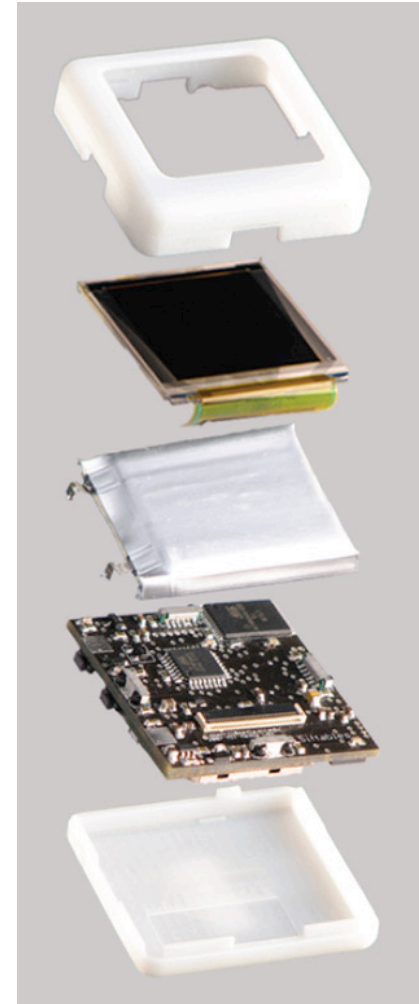
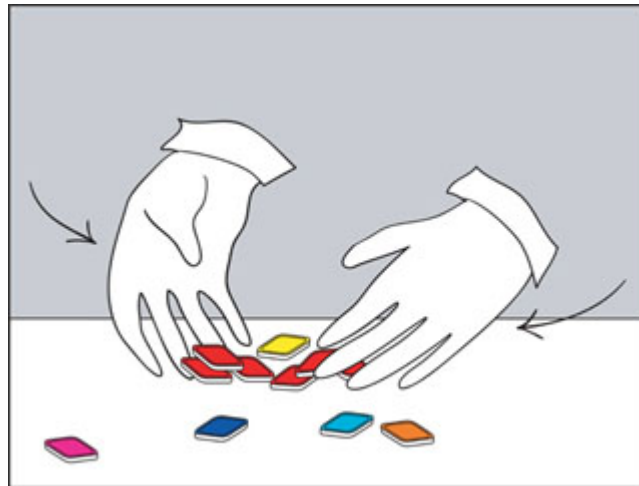
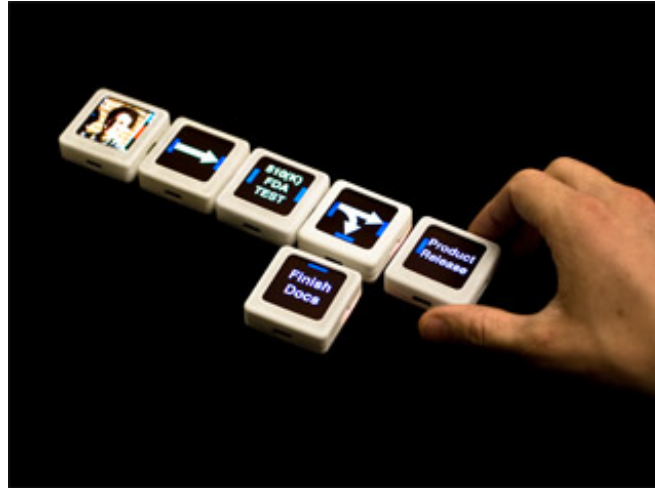
[Tangible Bits full paper presented at CHI 97](http://web.media.mit.edu/~ishii/)

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

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



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



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

Déjà une industrie !



iRobot® ConnectR™ Virtual Visiting Robot

Stay close to those you love – no matter where you are!

Don't miss out on special moments at home even when you are away. The iRobot ConnectR is a fun new way to see, talk to and interact with your loved ones, friends and pets – when you can't be there in person. Combining the latest in Internet communications and robot technology, ConnectR lets you virtually visit with loved ones, relatives and pets anytime you wish – seeing, hearing and interacting with them in their home as if you were there in person.



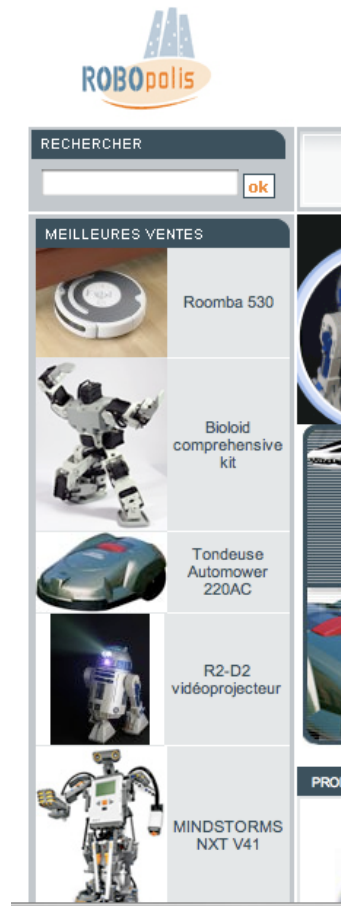
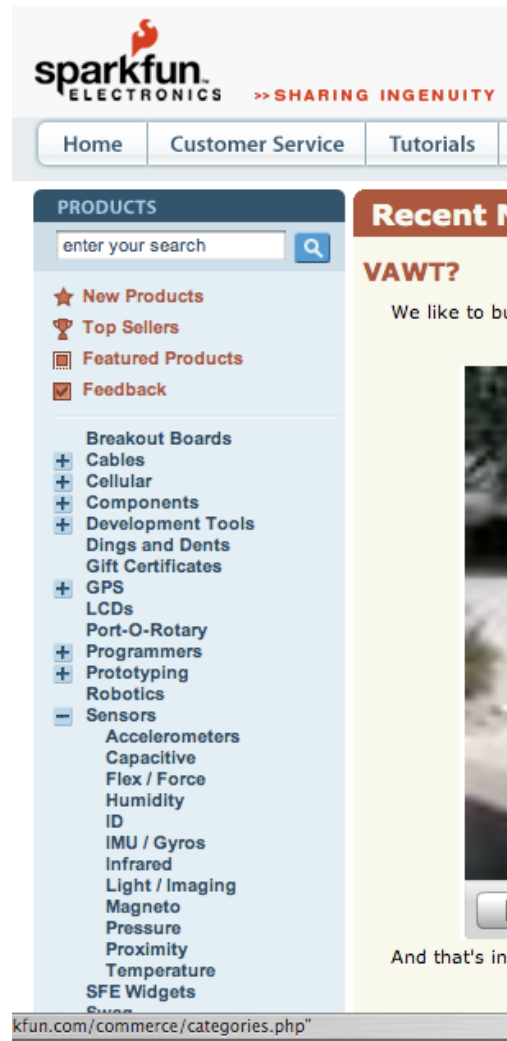
- Participate in family moments even though you're working late
- On a business trip? Read your kids a story and see their faces light up
- Join the fun from near or far
- Throw a party from a thousand miles away
- Tell Fido he's a "good boy" even while you're on vacation

IL BOUGE

skyscout

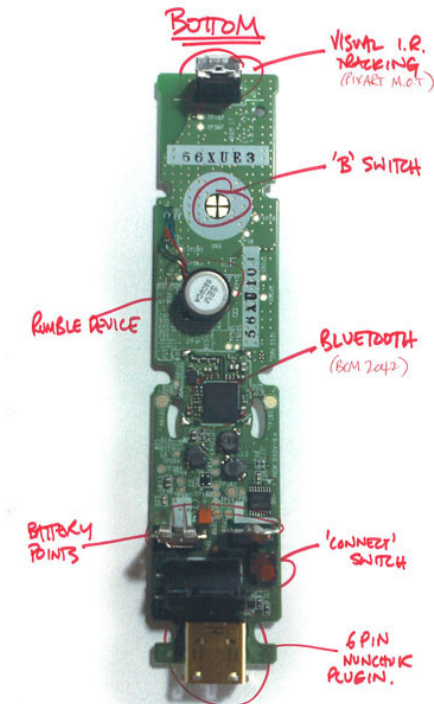
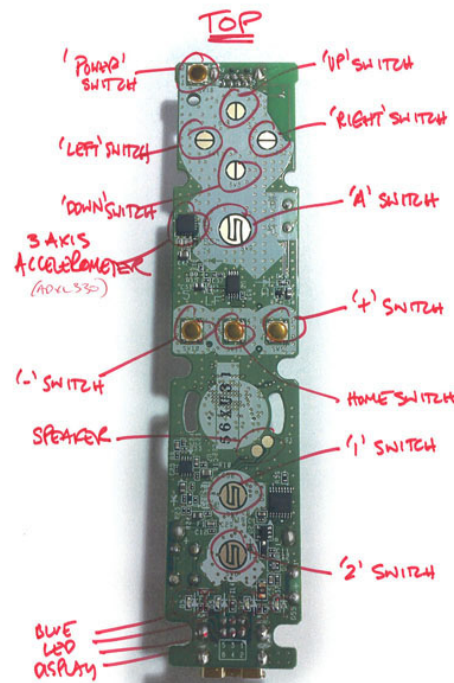


DIY (do it yourself) : les revendeurs



+ à Paris : StQuentin Radio, ACER, etc.

2. La Wiimote de Nintendo (2006)



Wiimote

Brett Rolfe, OneDigital

40 € wiimote + 20 €1

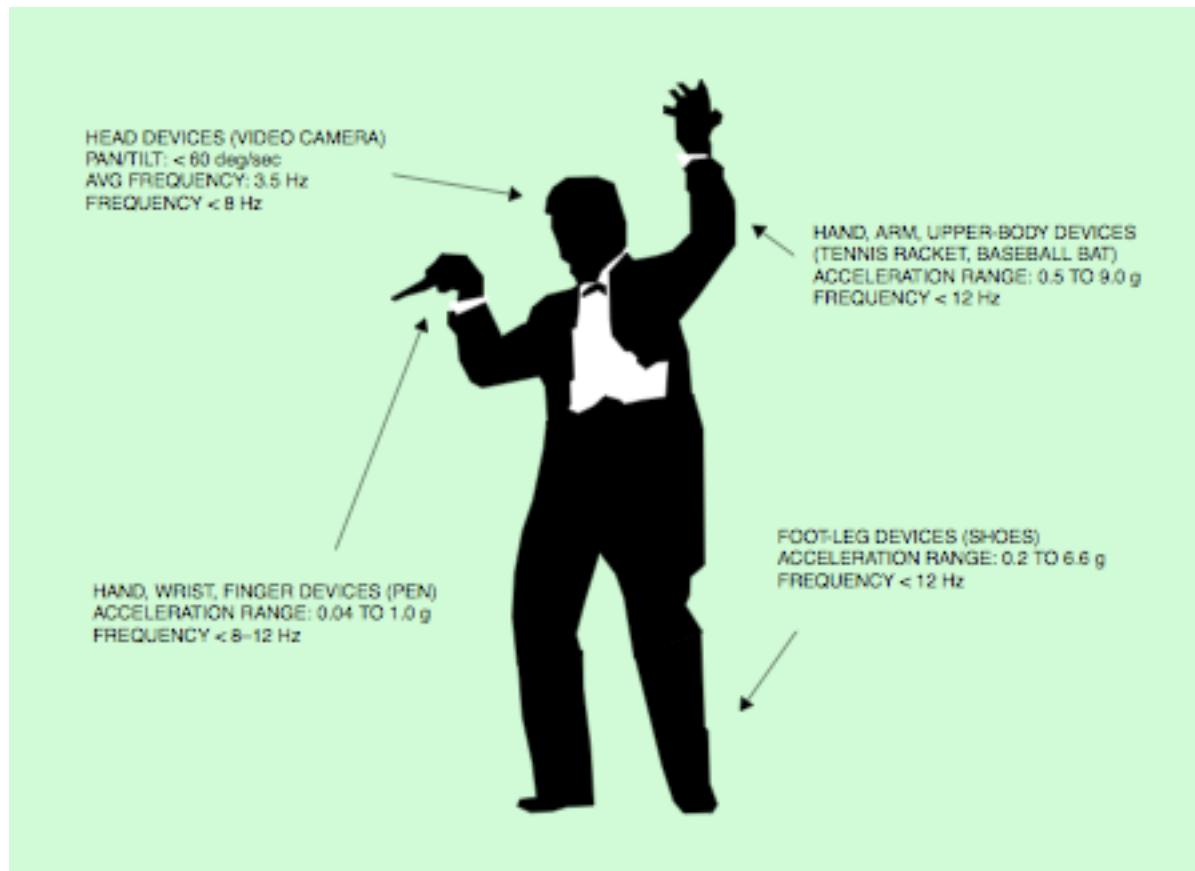
- accelerometre 3axes
- Camera IR + rec. Blobs
- HP, vibreur
- Plein de boutons + joysticks
- Bluetooth (et i2c avec le nunchuck)

Totalement « hacké »
=> www.wiili.com

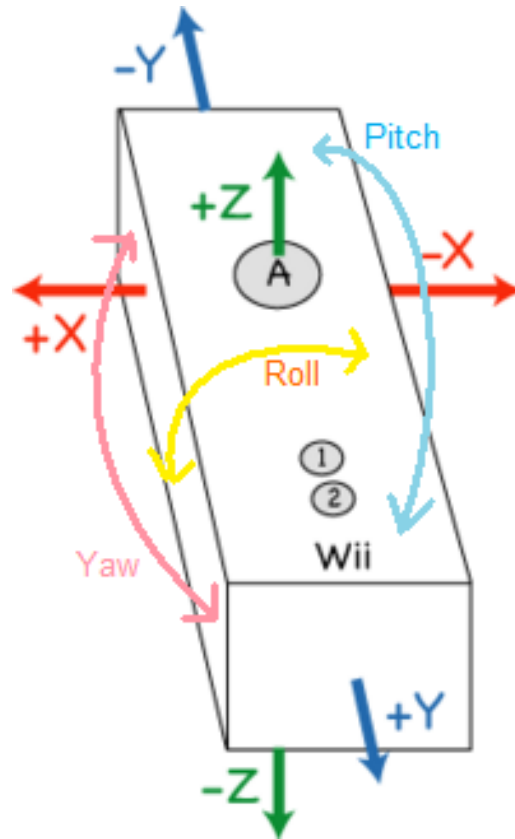
C. Verplaetse IBM Systems Journal 35(3-4) 1996 !!

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) :

$$ax = (xraw - x0)/(x3 - x0)$$

$$ay = (yraw - y0)/(y2 - y0)$$

$$az = (zraw - z0)/(z1 - z0)$$

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 coordonnées du vecteur gravité permet de d'obtenir l'orientation dans l'espace du dispositif en pitch et roll :

$$\text{pitch} = \arctan(ax / \sqrt{ay*ay + az*az})$$

$$\text{roll} = \arctan(ay / \sqrt{ax*ax + az*az})$$

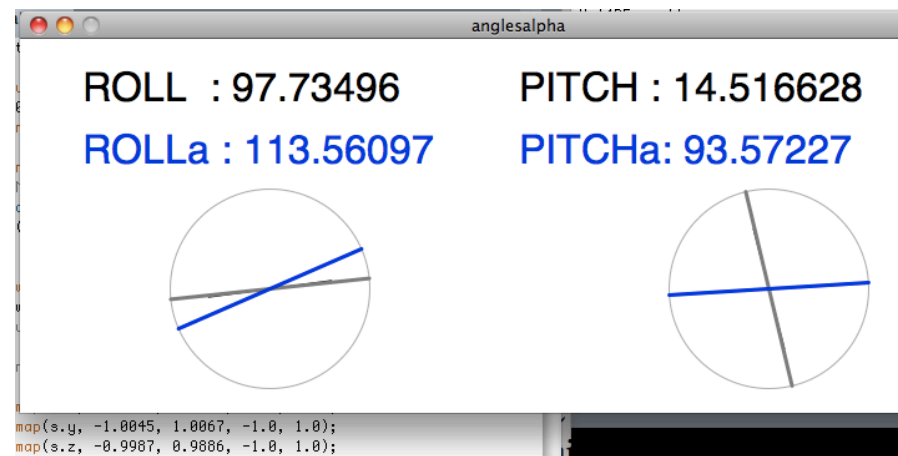
source : Kionix. *Tilt-sensing with Kionix MEMS Accelerometers*. Application note AN005. 2005 (en ligne sur www.kionix.com)

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

Un exemple d'utilisation avec Processing



Ma belle lunette Takahashi



Accéder à la wiimote
avec processing :
bibliothèque wrj4p5

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;
}
```

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 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)**

Calcul des angles avec filtrage des données



Boucle filtrage alpha très fort :

$$\text{estimé} = (1-\alpha) \cdot \text{estimé} + \alpha \cdot \text{mesure}$$

alpha fixé
a priori

Boucle de filtrage de Kalman :

$$K = (P+Q)/(P+Q+R)$$

$$\text{estimé} = (1-K) \cdot \text{estimé} + K \cdot \text{mesure}$$

$$P = (1-K) \cdot (P+Q)$$

R = écart type bruit

Q = écart type processus

Pinitial = 1

choix à faire entre réponse et stabilité

Le programme complet

```
import Ill.wrj4P5.*;
import Ill.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.isConnected()) return;
  // acquisition 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;

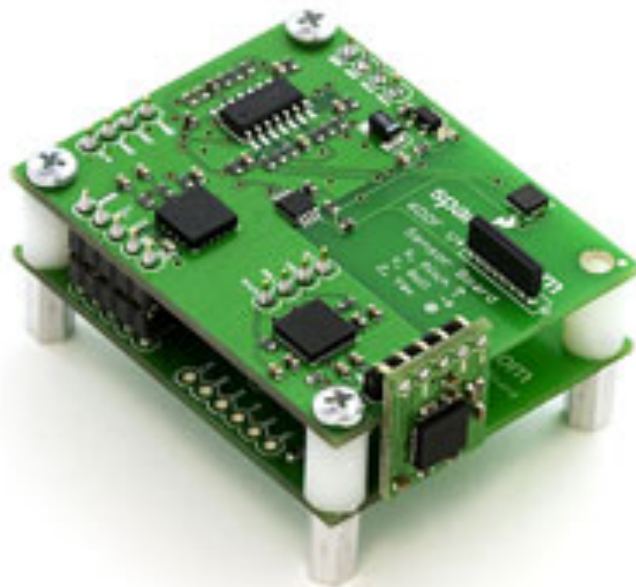
  // affiche les infos
  background(255);
  fill(0);
  text("ROLL : "+str(roll*180/PI), 50, 50);
  fill(0,0,225);
  text("ROLLa : "+str(rolla*180/PI), 50, 100);
  fill(0);
  text("PITCH : "+str(pitch*180/PI), 400, 50);
  fill(0,0,225);
  text("PITCHa : "+str(pitcha*180/PI), 400, 100);

  // 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();
}
```


Autres produits du même genre:



Sparkfun 450\$

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

Xsens



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



Utilisation de la caméra IR

// trouve sur <http://www.pobot.org/Camera-IR-avec-une-Wiimote.html>

```
import ll.wrj4P5.*;
import ll.Loc.*;
Wrj4P5 wii;

Loc[] irl;

void setup() {
  size(300,300);
  // connect first wiimote
  wii = new Wrj4P5(this).connect();

  smooth();
  frameRate(20); // sinon le wiimote ne peut pas suivre
}

void draw() {
  // wait until wiimote connected
  if (wii.isConnecting()) return;
  Loc s = wii.rimokon.sensed;

  background(0);

  wii.rimokon.setLED(1,true);
  //wii.rimokon.setLED(2,false);

  irl = wii.rimokon.irLights;
  fill(255,0,0);
  ellipse(irl[0].x*width,irl[0].y*height,20,20);
  fill(0,255,0);
  ellipse(irl[1].x*width,irl[1].y*height,20,20);
  fill(0,0,255);
  ellipse(irl[2].x*width,irl[2].y*height,20,20);

  println(irl[0].x);
}
```

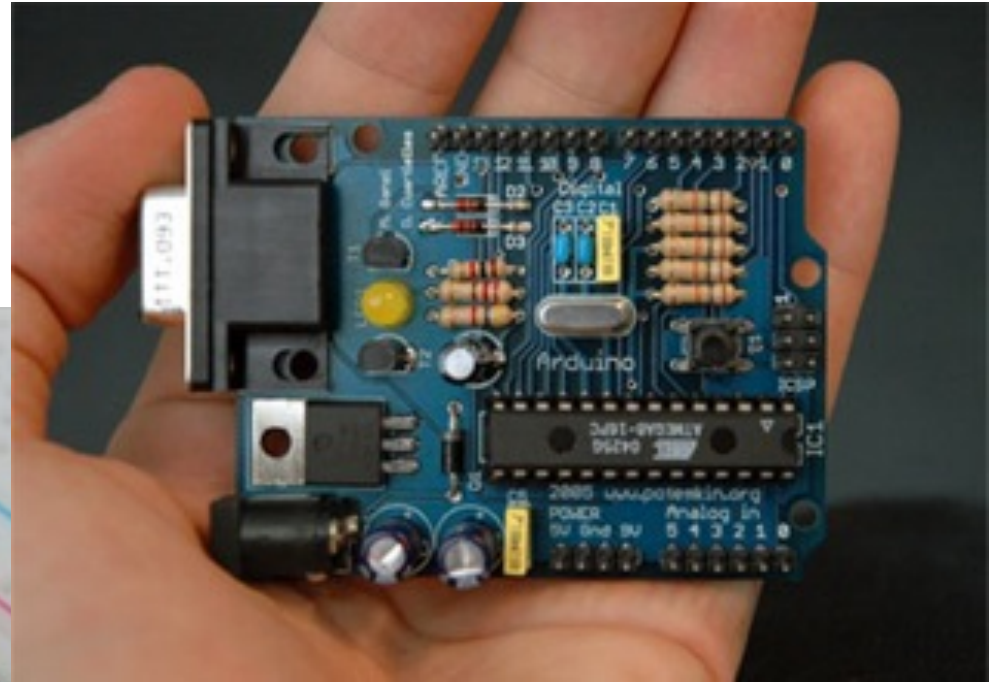
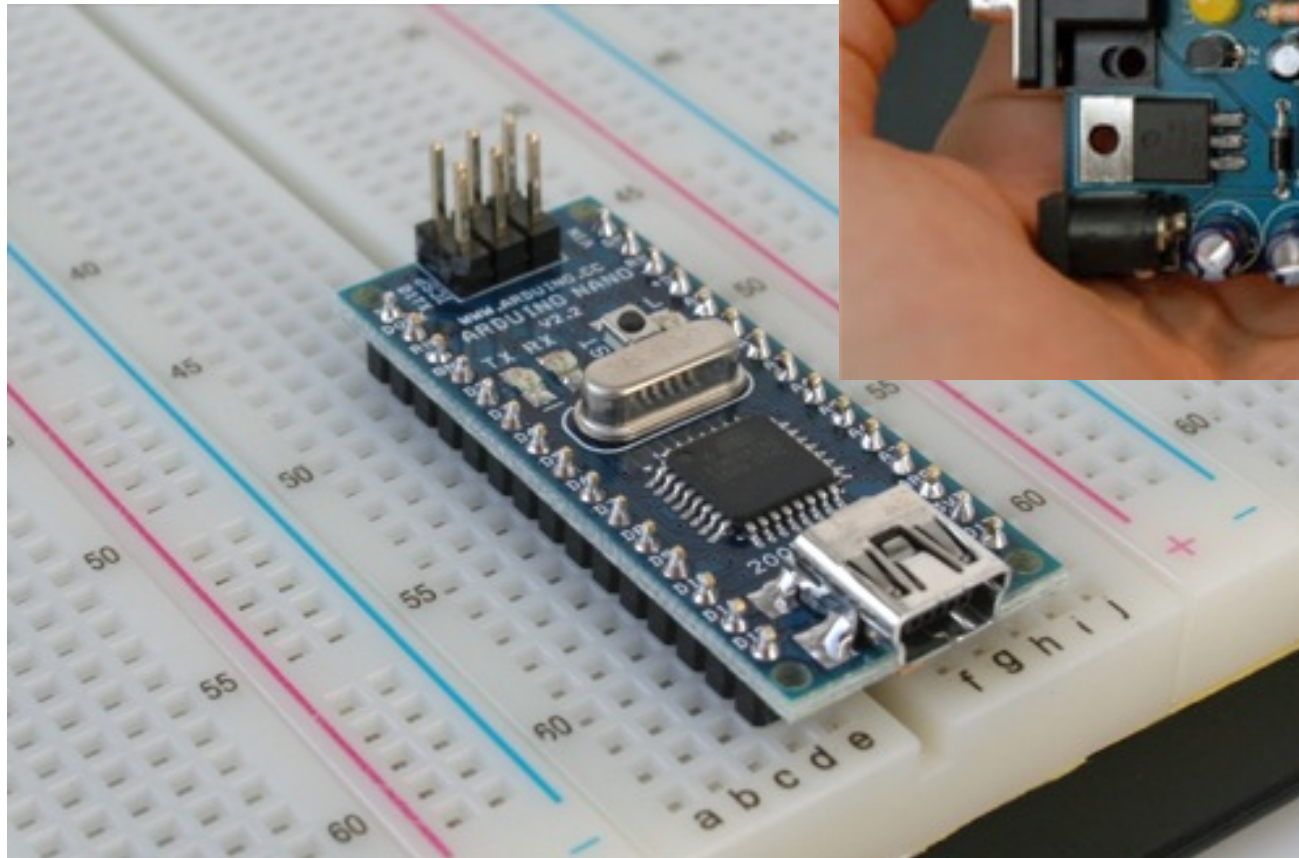


3. La carte ARDUINO



(Science et vie junior - février 2012)

www.arduino.cc



+ une version
bluetooth

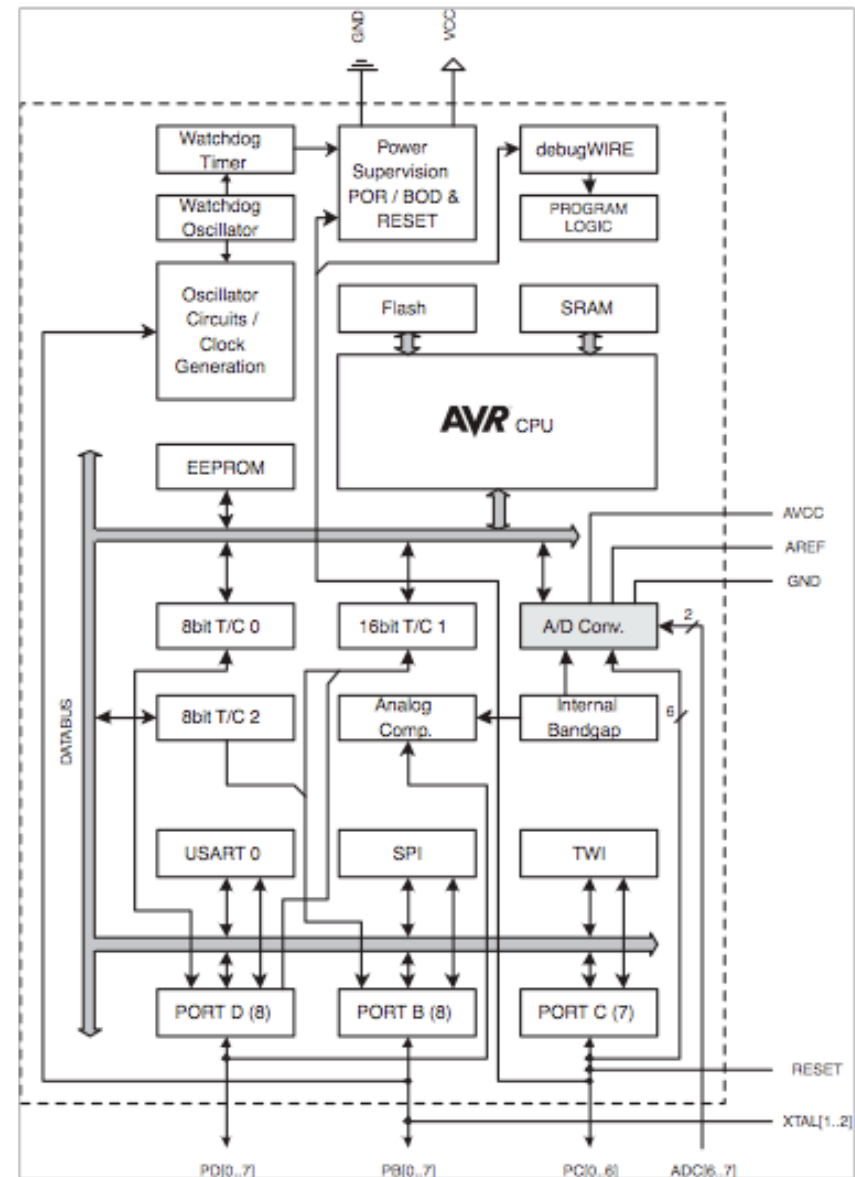
En France : AlyaSoft (à Béziers) : 22 euros + lextronic + st quentin radio

Le μ contrôleurs 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

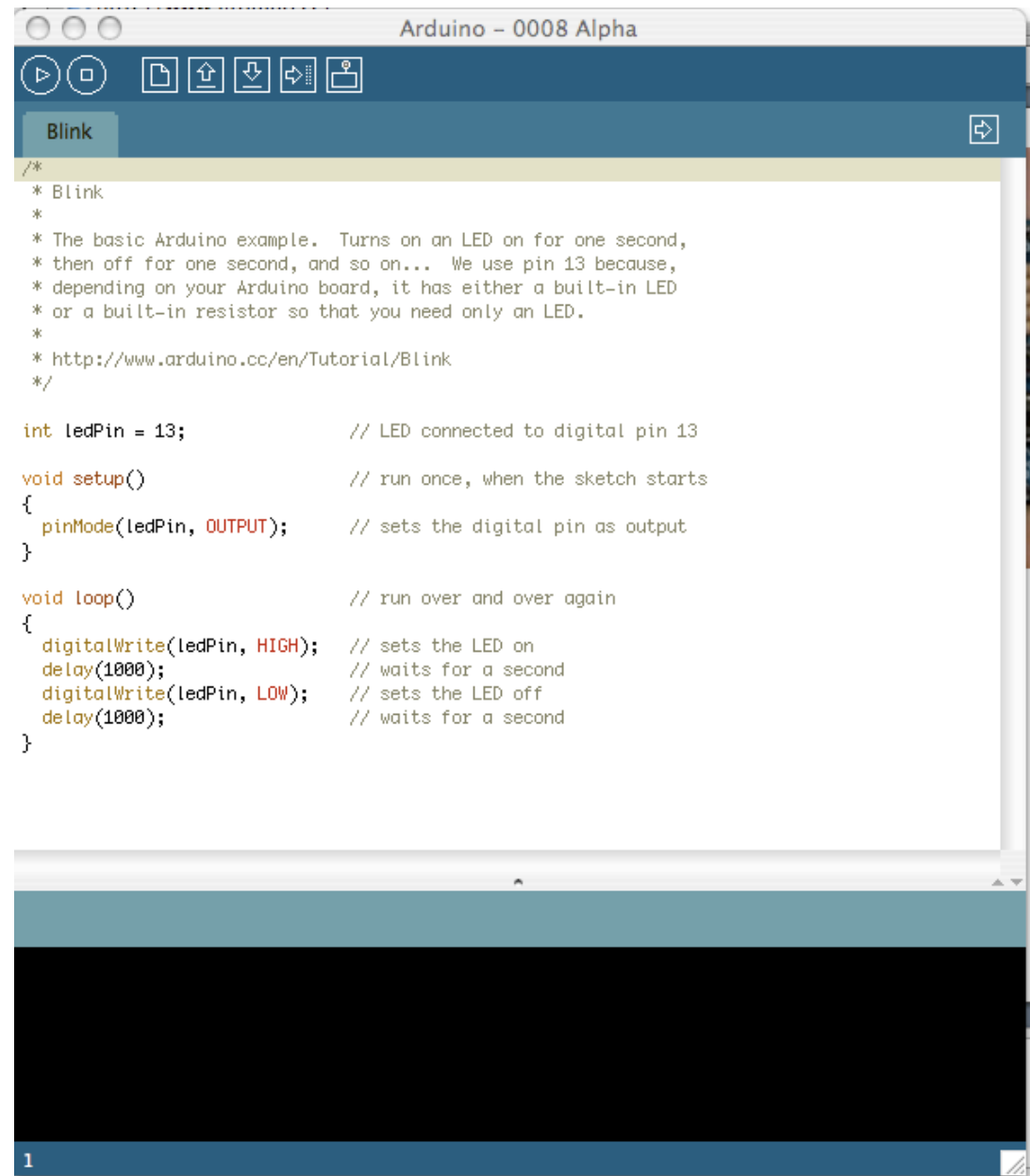
Multiplateforme !
Open source !

Basé sur C/C++

Types flottants !!
fcts math
tableaux

Librairies de com
Serie, i2c (Wire)

Contrôle servos ...

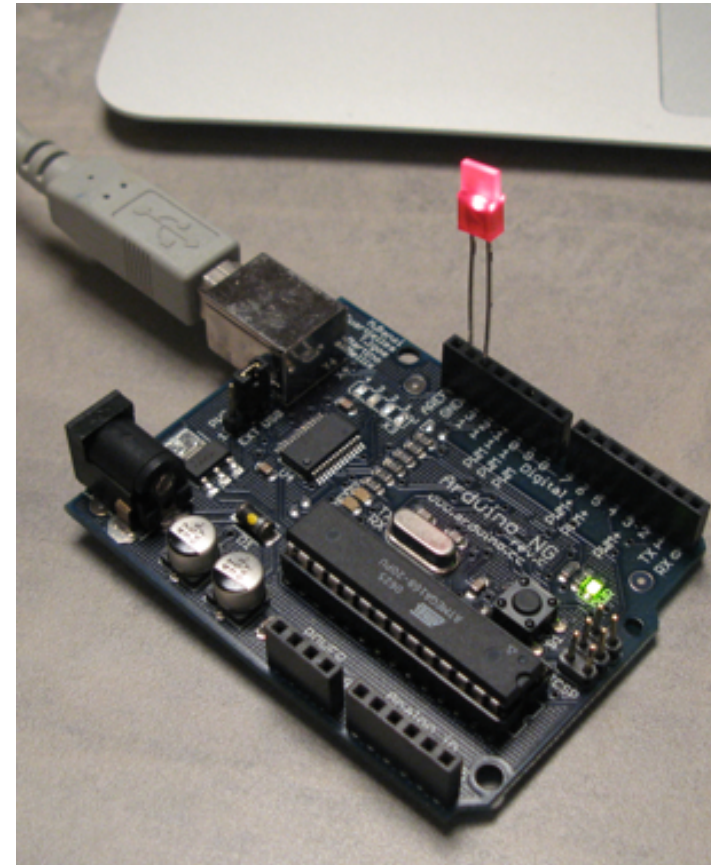
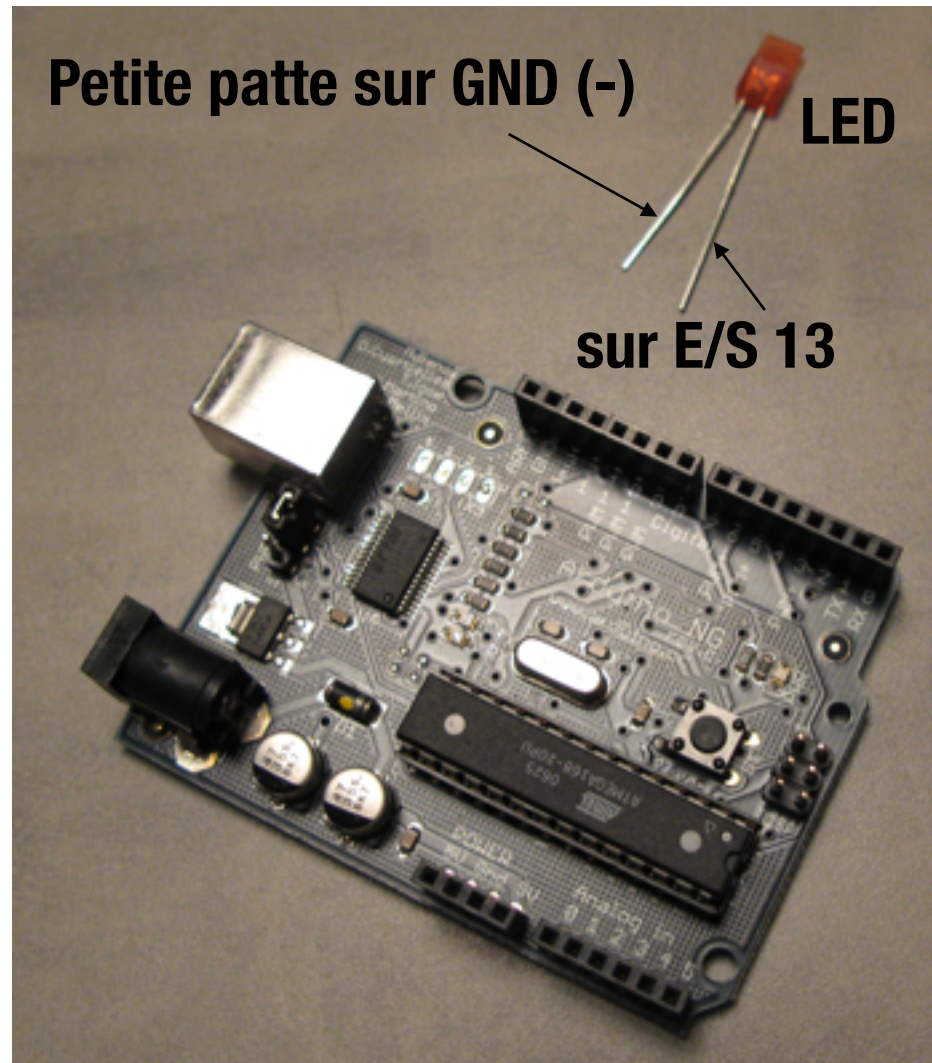
A screenshot of the Arduino IDE window titled "Arduino - 0008 Alpha". The window shows the "Blink" sketch. The code is as follows:

```
/*  
 * Blink  
 *  
 * 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  
}
```

The IDE interface includes a toolbar with icons for running, saving, and other functions. The status bar at the bottom shows the line number "1".

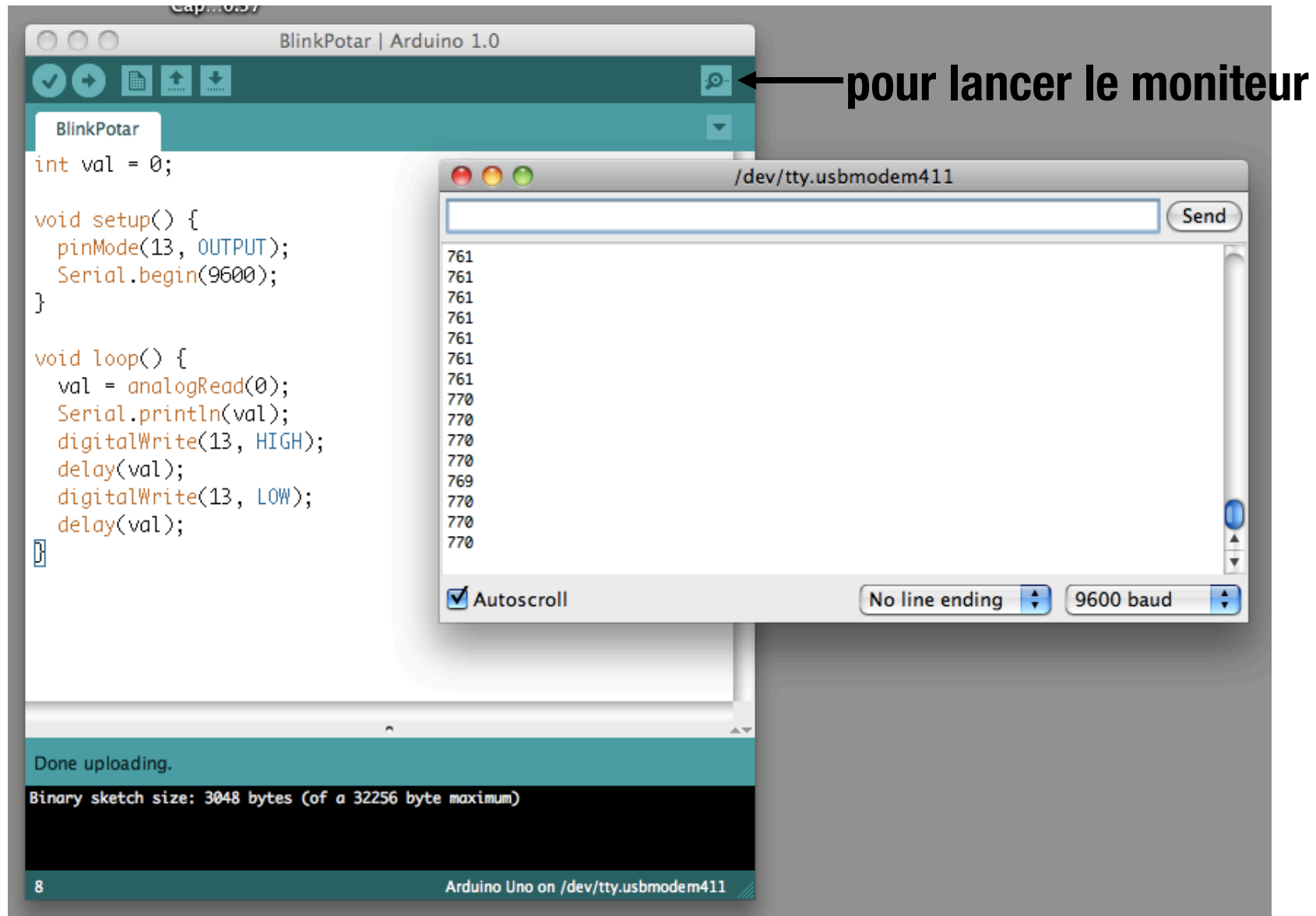
« blink » : le Hello world de l'Arduino

Blink : le montage

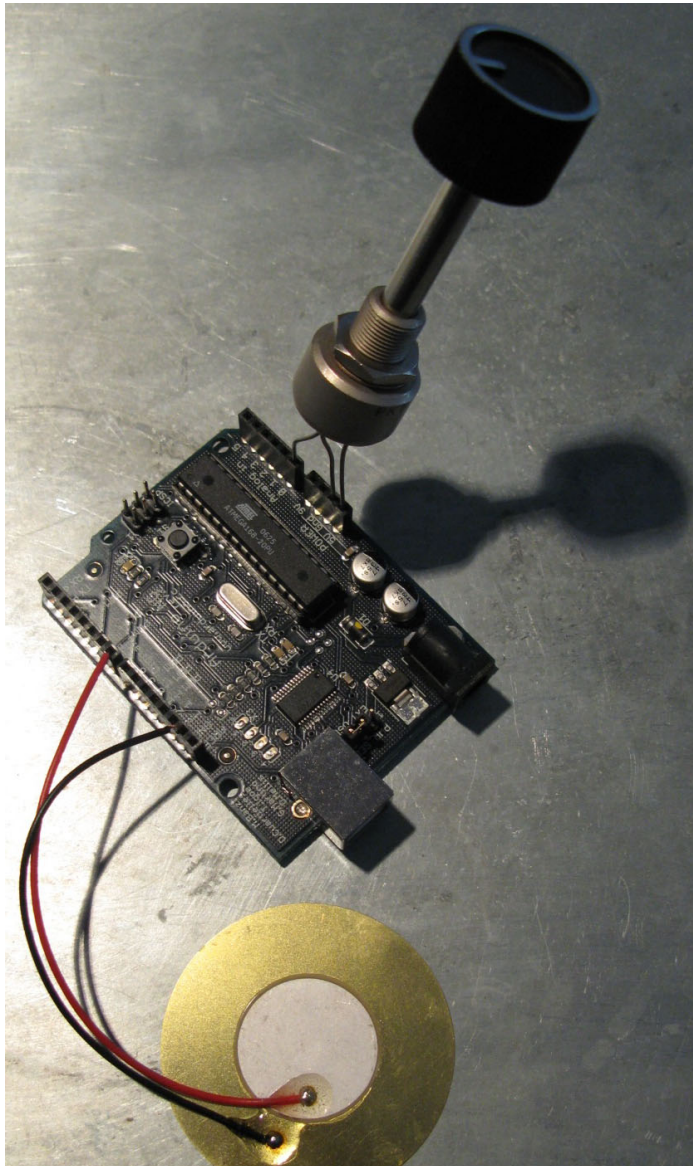


Fiat lux ...

Entrée analogique (et liaison série)



Buzzer :



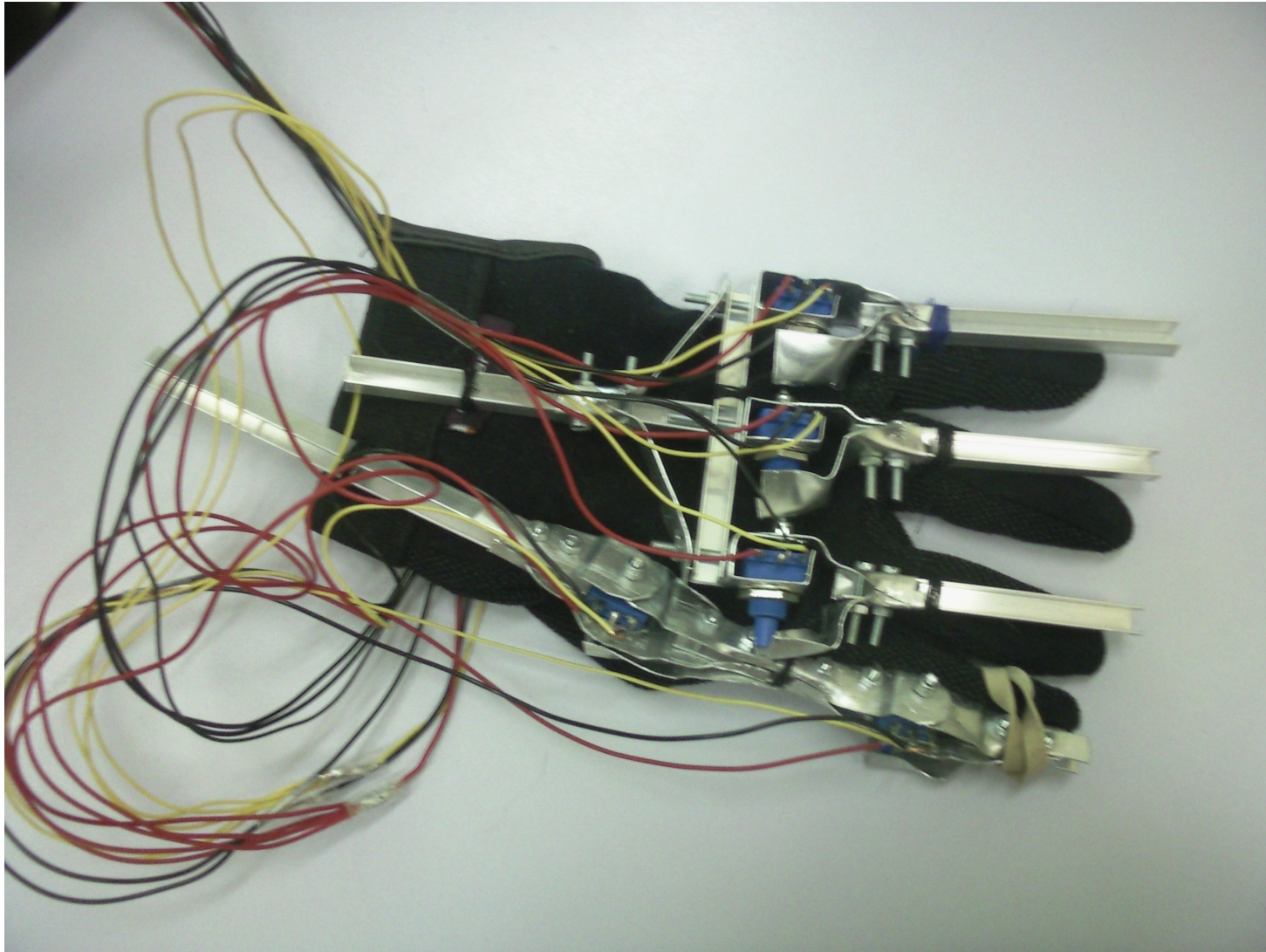
```
/* copleft 2006 Tod E. Kurt <tod@todbot.com  
* http://todbot.com/  
*/
```

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

```
int val = 0;
```

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

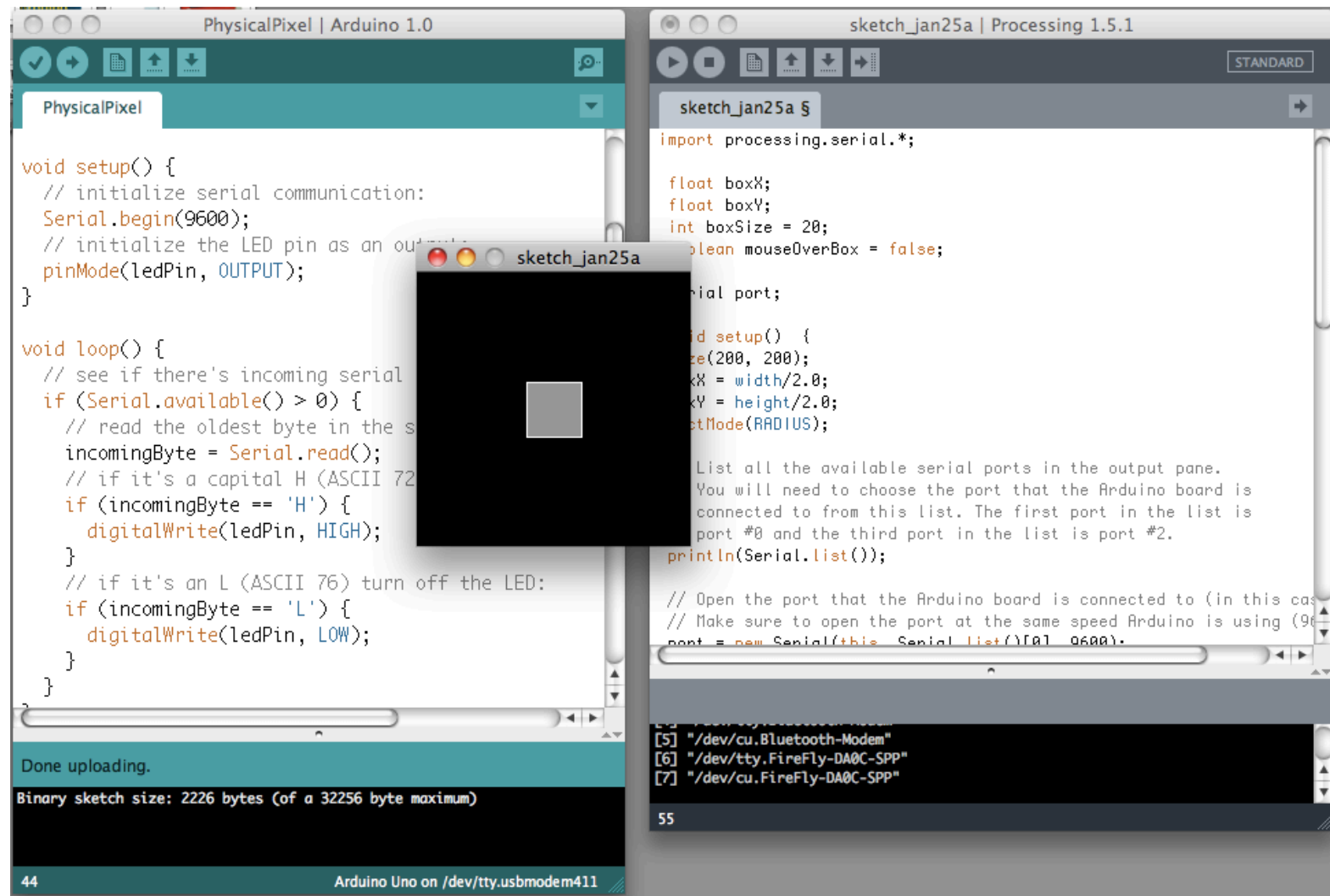
```
void loop() {  
  digitalWrite(speakerPin, LOW);  
  val = analogRead(potPin);  // read value from the sensor  
  val = val*2;               // process the value a little  
  for( int i=0; i<500; i++ ) { // play it for 50 cycles  
    digitalWrite(speakerPin, HIGH);  
    delayMicroseconds(val);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

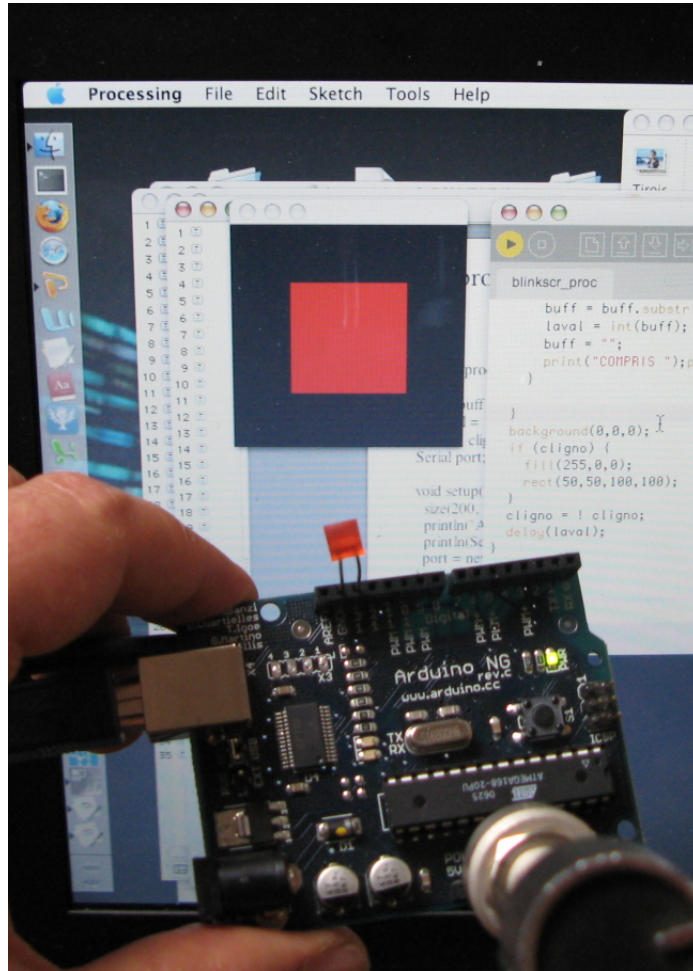
gant de captation (élève ingénieur cnam paris, 2010)

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;

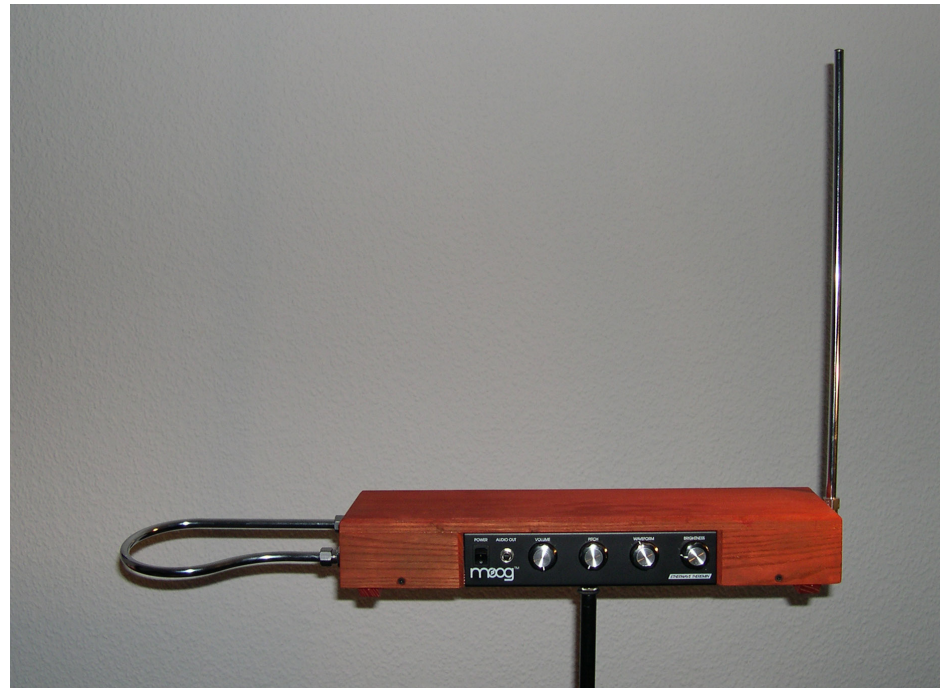
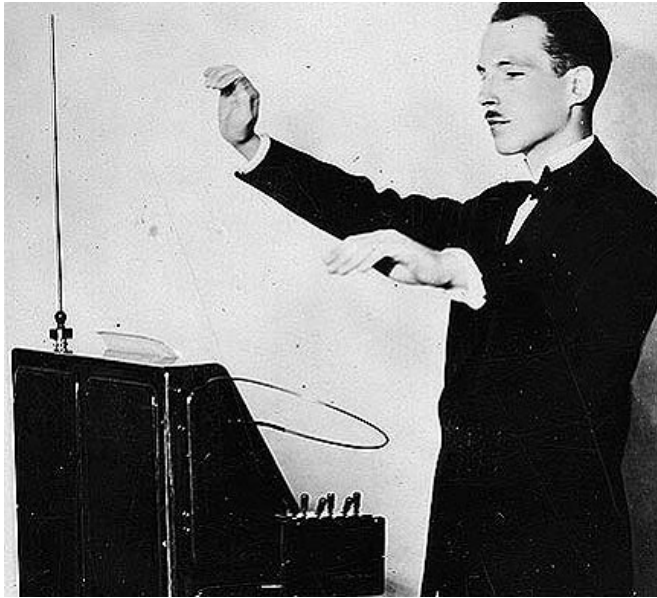
void setup() {
  size(200, 200);
  println("Available serial ports:");
  println(Serial.list());
  port = new Serial(this, "/dev/tty.usbserial-A50018tg", 9600);
}
```

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

Les deux processus synchrones ???

```
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);
}
```

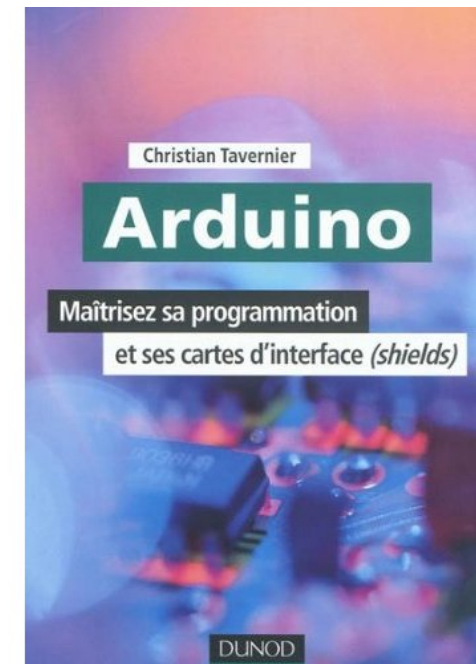
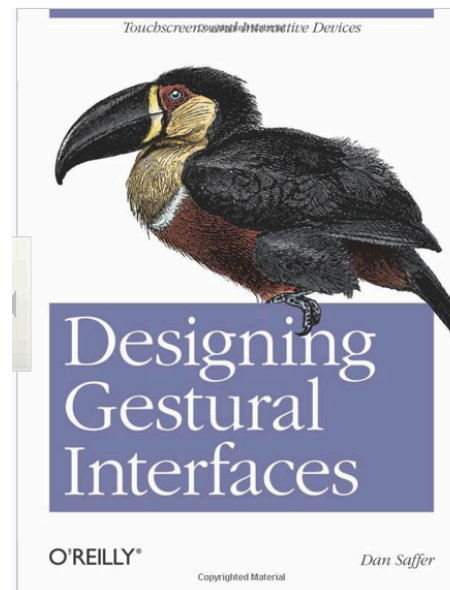
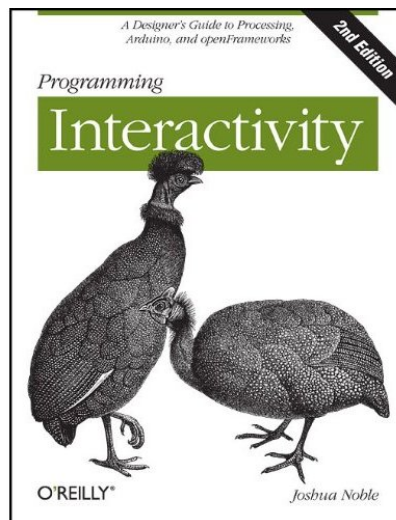
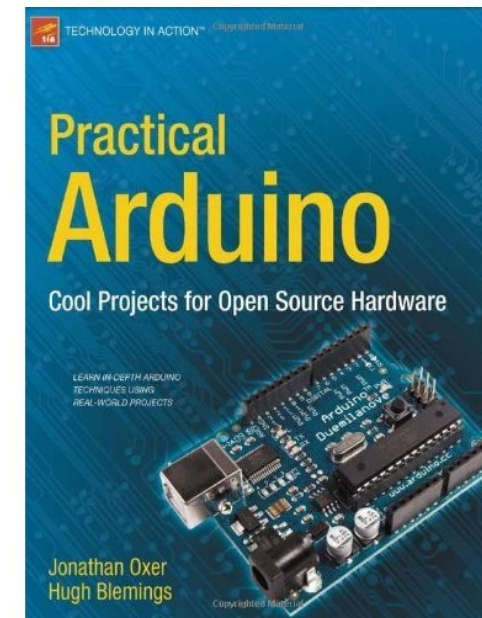
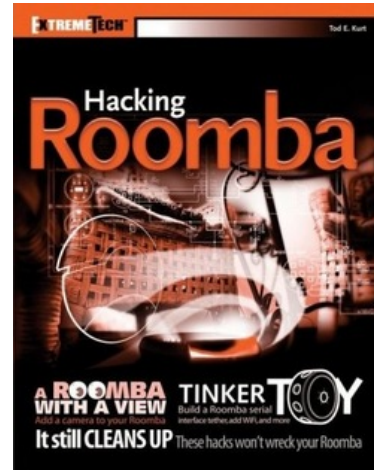

Un bon projet : (re)faire un "Theremin"



Pour continuer :

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

Bibliographie



Et un site (parmi plein)

