

Introduction au Physical Computing: “Faire parler les choses”

Didier Donsez, Olivier Richard

Polytech' Grenoble/UJF/INPG/CNRS/INRIA

6 et 7 Julliet , 2011, ETIA @ Lille

- Physical Computing
- Arduino
- Rappels Electronique
- Divers

Emprunts: Des sites Arduino, Wikipedia, Fritzing et bien d'autres...

Licence: Creative Commons Attribution-ShareAlike 3.0 License.¹

¹<http://creativecommons.org/licenses/by-sa/3.0/>

Physical Computing: Au sens large, construire des **systèmes physiques** interactifs qui utilisent **des logiciels** et du matériel pouvant s'interfacer avec des **capteurs** et des **actuateurs**.

- Cultures: DIY (Do-It-Yourself), Hacker, Arts, Domaine de l'embarqué
- **Informatique et électronique**
- Microcontrolleurs
- Capteurs, actuateurs
- Prototypages
- Education

Physical Computing: exemple

Exemple: Les mécaniques poétiques d'EZ3kiel



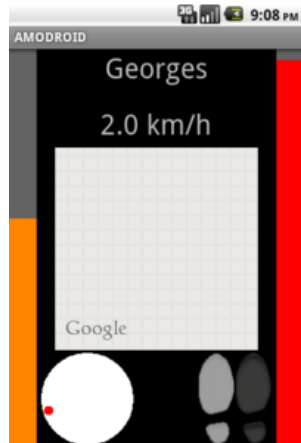
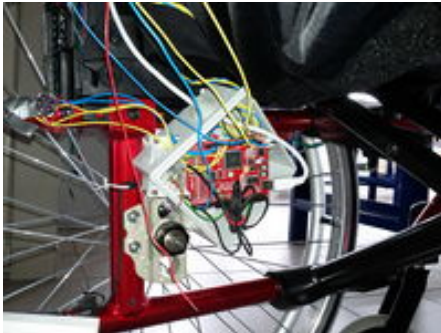
Projets de l'école Polytech' Grenoble

● *Intelligent and communicating wheelchair*



Physical Computing: exemple

- *Intelligent and communicating wheelchair*





Arduino² : **Simplicité**

- *Une carte avec microcontrôleur (autonome)*
- *Un langage*
- *Un environnement de développement intégré*
- *Mise en route en **15 min maximum***

Ce projet est basé sur les projets *Processing*³ et *Wiring*⁴

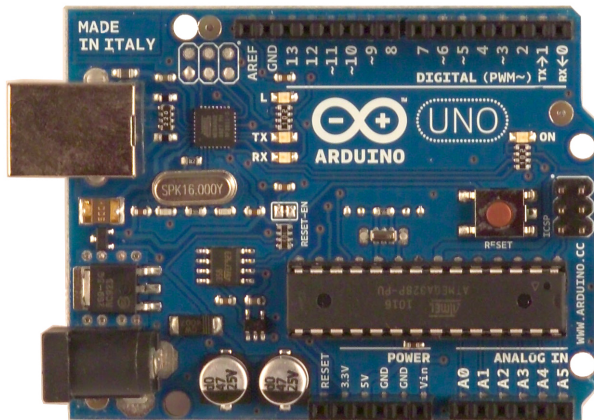
²<http://www.arduino.cc/>

³<http://wiring.org.co/>

⁴<http://www.processing.org/>

La carte Arduino

Prix 22-25€

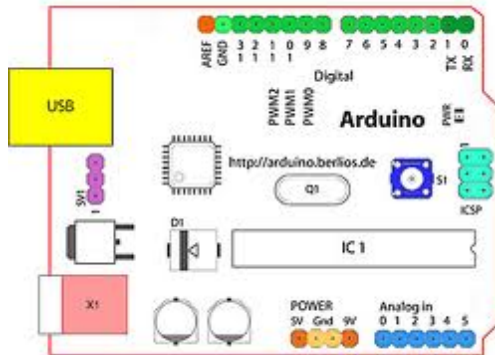


**USB: Pour l'alimenter, la programmer et communiquer
(communication série sur USB)**

Le Microcontrôleur de l'Arduino (UNO) : le cœur

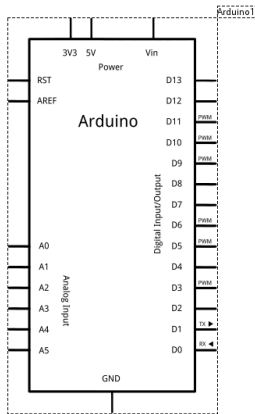
| | |
|-------------------------|------------------------------------|
| Microcontroller | ATmega328 (Atmel) |
| Operating Voltage | 5V |
| Supply Voltage | 7-12V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (0.5 KB used by bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Clock Speed | 16 MHz |

La carte Arduino: vue plus schématique



Pins: pattes. On parle plutôt de **port d'entrées/sorties** digital, analogique. Les ports sont généralement paramétrable, le plus simple le mode: soit entrée, soit sortie. :width: 60%

La carte Arduino: vue schématique



Made with  Fritzing.org

Une présentation simplifiée de C/C++. Pour une première prise en main rapide

```
void setup() {  
    // initialize the digital pin as an output.  
    // Pin 13 has an LED connected on most Arduino boards:  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH); // set the LED on  
    delay(1000); // wait for a second  
    digitalWrite(13, LOW); // set the LED off  
    delay(1000); // wait for a second  
}
```

Arduino : syntaxe du langage (1/2)

Voir la section reference du site Arduino⁵

Program structure

```
void setup() { ... } void loop() { ... }
```

Statements

```
; {}
```

```
// /* */
```

```
#define #include
```

Control structures

if, if...else, for, switch case, while, do... while

break, continue, return, goto

Control structures

```
type func(type param, ...)
```

⁵<http://www.arduino.cc/reference>

Arduino : syntaxe du langage (2/2)

Data Types

void, boolean, char, unsigned char, byte,
int, unsigned int, word, long, unsigned long
float, double, string (char[])

Constructor

[]

Variables

local, global, static local, volatile, const, sizeof()

Operators

Arithmetic = + - * / %

Comparison == != < > <= >=

Boolean && || !

Bitwise & | ^ ~ « »

Pointer Access * &

Compound == -- += -= *= /= &= |=

IDE Arduino: l'environnement de développement intégré

- Edition de **sketch**
- Compilation
- Chargement du code sur la carte
- Console ligne série sur USB
- Base d'exemple
- Librairie
- **Sélection le bon port de la ligne série**
- **Choisir la bonne carte**



The screenshot shows the Arduino IDE interface. The title bar reads "Arduino - 0011 Alpha". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, saving, opening, and other functions. The main text area displays the "Blink" sketch, which is a standard Arduino example for controlling an LED. The code is as follows:

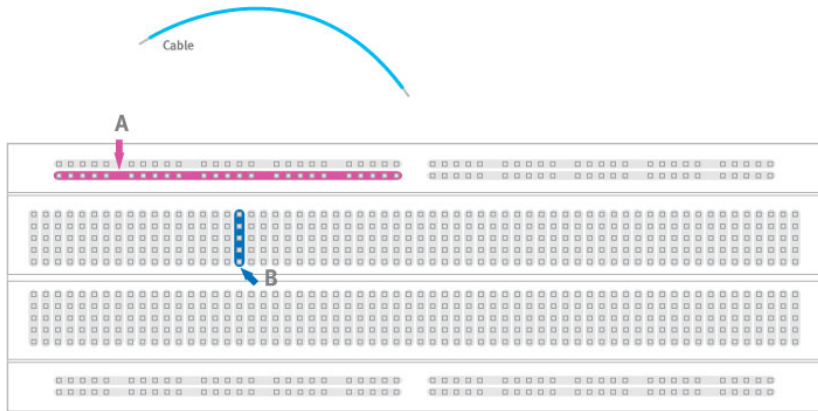
```
/*  
 * Blink  
 *  
 * The basic Arduino example. Turns on an LED on for one second,  
 * then off for one second, and so on... We use pin 13 because,  
 * depending on your Arduino board, it has either a built-in LED  
 * or a built-in resistor so that you need only an LED.  
 *  
 * http://www.arduino.cc/en/Tutorial/Blink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

```

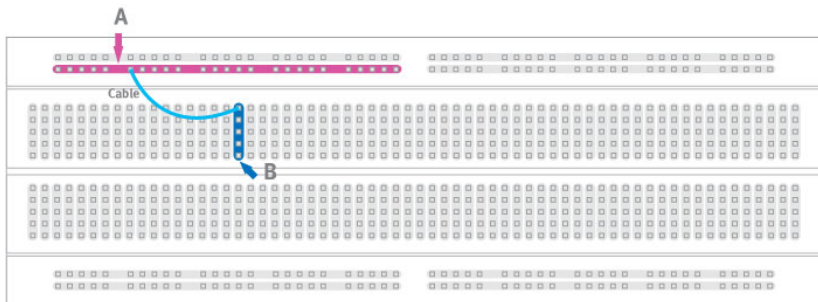
At the bottom of the IDE, there is a status bar showing "Done compiling." and "Binary sketch size: 1098 bytes (of a 14336 byte maximum)". The line number 22 is visible at the bottom left of the code area.

Plaque d'essai: Breadboard

- Élément essentiel pour le prototypage et essai en tout genre

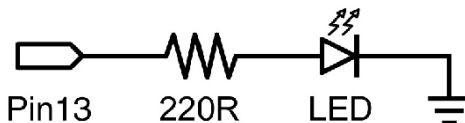


Plaque d'essai: Breadboard



Led et résistance

- **Attention**, on ne branche pas n'importe quoi n'importe comment.

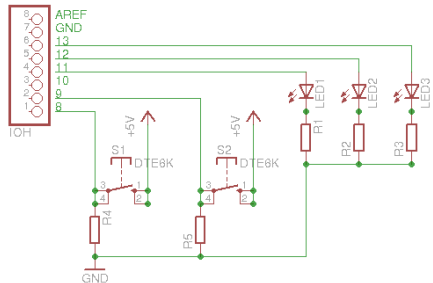


- **Pas** de moteur/lampe, de composants nécessitant de la puissance directement connecté à un port d'entrée/sortie qui est **en mode sortie** !

Bouton et résistance

- Résistances de *Pull Up* ou *Pull Down* (tirage vers le haut/bas), pour la valeur par défaut sinon on a une antenne et donc du bruit

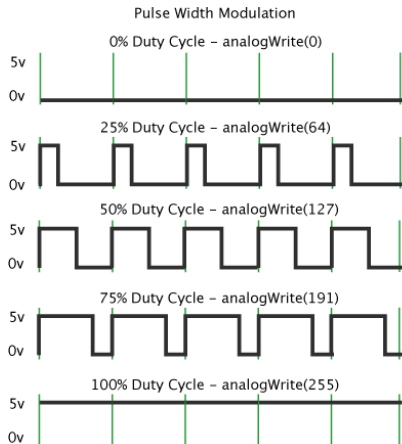
```
void setup() {  
    pinMode(8, INPUT)  
    pinMode(9, INPUT)  
}  
  
void loop() {  
    int s1 = 0, s2 = 0;  
    while (!s1 && !s2) {  
        s1=digitalRead(8);  
        s2=digitalRead(9);  
    }  
    if (s1 && !s2) {...} else  
    if(!s1 && s2) { }  
}
```



PWM : Modulation en large d'impulsion

PWM: Pulse Wide Modulation Duty cycle = rapport cyclique

Utilisation: commande de servo moteur, fading pour led



- **Attention** : entre 0v et 30v pas de risque, à 220V c'est mortel
- **Les manipulation à 110v ou 220v** : demande un minimum de connaissance et une rigueur stricte (demandez conseil à une personne expérimentée)
- Consultez les schémas et les programmes sur les sites⁶ et⁷ (très nombreux exemples)
- Pour des simulations de la partie électronique utilisez des simulateurs et expérimentez : icircuit⁸ ou Ktechlab⁹

⁶<http://www.arduino.cc/>

⁷<http://www.fritzing.org/>

⁸<http://www.falstad.com/circuit/>

⁹<http://sourceforge.net/projects/ktechlab/>

Arduino: Interruptions

- Motivation : avoid polling (with complex timing calibration)
- External interruptions
 - Digital pin 2 and 3 on Arduino¹⁰
 - `attachInterrupt(interrupt, funct, mode)`, `detachInterrupt(funct)`
 - **mode** = LOW, CHANGE, RISING, FALLING
- Critical section
 - `noInterrupts(); ... interrupts();`

¹⁰+ digital pin 21, 20,19,18 on Mega (version musclée de l'Arduino)

Arduino complément: Interruptions

- Example

```
#define LED 13;
volatile int state = LOW;
void setup() {
    pinMode(LED, OUTPUT);
    attachInterrupt(0, blink, CHANGE);
}
void loop() { digitalWrite(LED, state); }
void blink() { state = !state; } //invert state
```


Arduino: Shields

- Extension de pour l'Arduino
- Principaux:
 - Bluetooth
 - Zigbee
 - GPS
 - Plaque d'essai
 - Commande de moteur
 - ...



- Variante
 - *Lilypad*
 - *Paperduino*
- Vers le 32 bit
 - *ARM 32bits Cortex-M3: Leaf Maple et*
 - *PIC 32: chipKIT Uno32*
- Ersatz
 - *Netduino (C#)*
 - *Fez*

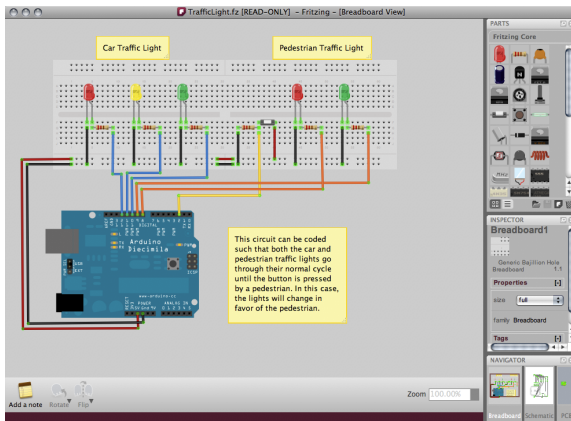
- Interactivity programming
- Program structure

```
void setup() { ... }
```

```
void draw() { ... }
```

- Code snippets
 - *2D, 3D and OpenGL*
 - *Data (File I/O, Web APIs, Datatypes)*
 - *Images*
 - *Interaction and Interface*
 - *Math and Physics*
 - *Inter-app Communication*
 - *Serial with Arduino(s)*
 - *Export as a application or a Applet*
- Processing.js for Javascript-based RIA


- Pour dessiner des dessins de type breadboard, **des schémas**, des PCB (typon) en vue de réaliser des circuits.
- Le site¹¹ regorge de schémas.



¹¹<http://www.fritzing.org/>

Arduino: des émules

Une salle¹² dans l'esprit des fablab pour les projets des étudiants de Polytech' Grenoble

¹²<http://air.imag.fr/mediawiki/index.php> 

- Physical Computing: contrôler, mesurer, automatiser **simplement**
- Ardiuno: la plate-forme complète référence
- Simplicité de mise en oeuvre
- TP simple à mettre en place (ca tient dans une valise)
- Atout pédagogique: de l'idée à la concrétisation
- Une communauté très actives
- Une bonne entrée pour du hacking plus poussé
- Faible coût
- Limité en puissance: plusieurs Arduino ou basculer sur de l'ARM 32bit Cortex-M3
- Il est l'heure de manipuler: http://air.imag.fr/mediawiki/index.php/Travaux_Pratiques_ETIA

- Tom Igoe, Making Things Talk: **Practical Methods for Connecting Physical Objects**, Make Books , 2007, ISBN-10: 0596510519
- Banzi, Massimo (March 24, 2009). **Getting Started with Arduino (1st ed.)**. Make Books. pp. 128. ISBN 0596155514.
<http://www.makershed.com/ProductDetails.asp?ProductCode=9780596155513>.
- Ozer, Jonathan; Blemings, Hugh (December 28, 2009). **Practical Arduino: Cool Projects for Open Source Hardware (1st ed.)**. Apress. pp 500. ISBN 1430224770. <http://www.apress.com/book/view/9781430224778>.
- Noble, Joshua (July 15, 2009). **Programming Interactivity: A Designer's Guide to Processing, Arduino, and openFrameworks (1st ed.)**. O'Reilly Media. pp. 768. ISBN 0596154143.
<http://oreilly.com/catalog/9780596800581/>.

- Schmidt, Maik (November 20, 2010). **Arduino: A Quick-Start Guide (1st ed.)**. The Pragmatic Bookshelf. pp. 275. ISBN 978-1-93435-666-1.
<http://pragprog.com/titles/msard/arduino>.
- Sparkfun Inventor'Guide
 - <http://www.sparkfun.com/tutorial/AIK/CIRC00-sheet-SPAR.pdf>
 - <http://www.sparkfun.com/tutorial/AIK/ARDX-EG-SPAR-PRINT-85.pdf>
- Lextronic, Gotronic, DealExtreme