

# TP microcontrôleur sur carte Arduino

Nicolas Palix, Pierre-Louis Aublin et Didier Donsez

6-8 juin 2011

**Note** : L'objectif de cette séance est de vous faire découvrir les microcontrôleurs dans le cadre d'un usage "ludique". Pour des raisons techniques, il existe plusieurs projets utilisant différents périphériques mais tous ont en commun l'utilisation de carte de prototypage Arduino Uno. Bonne séance!

## 1 Ressources

Lors de cette séance de travaux pratiques, vous utiliserez les ressources Internet liées à votre sujet ainsi que l'environnement de développement **Arduino** (disponible sur les machines) et l'environnement **Processing** (pour certains sujets uniquement, également préinstallé).

### 1.1 Serveur proxy

Pour accéder à Internet, il peut être nécessaire de configurer le serveur mandataire (*proxy*) utilisé par votre navigateur. Utilisez la configuration de serveur mandataire suivante : `www-cache:3128`

### 1.2 Arduino

Website : <http://www.arduino.cc/>

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

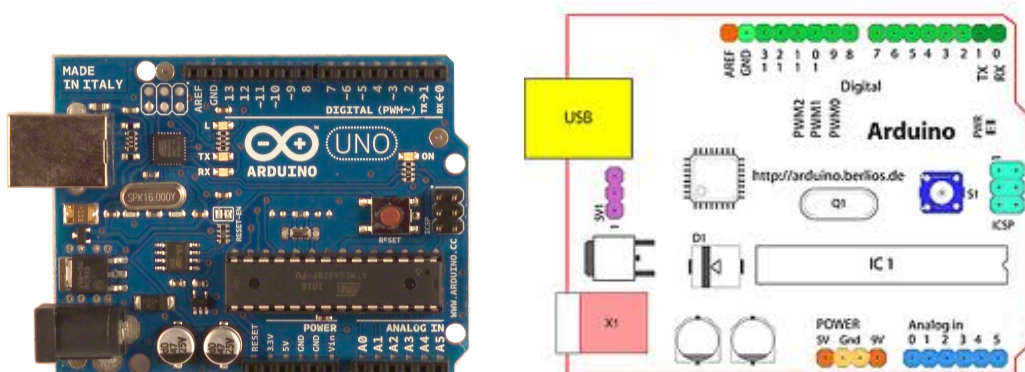


FIGURE 1 – Carte Arduino

### 1.3 Processing

Website : <http://www.processing.org/>

Processing is an open source programming language and environment for people who want to create images, animations, and interactions. Initially developed to serve as a software sketchbook and to teach fundamentals of computer programming within a visual context, Processing also has evolved into a tool for generating finished professional work. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

## 2 Sujets des projets

La description complète des sujets est disponible à l'adresse suivante :  
[http://air.imag.fr/mediawiki/index.php/Stage\\_PEIP](http://air.imag.fr/mediawiki/index.php/Stage_PEIP)



### 3 Documentation

L'utilisation de la liaison série est illustrée par les deux exemples suivants. Le premier montre son utilisation afin d'émettre des informations depuis l'Arduino. Le second exemple illustre la réception d'information avec Processing.

#### 3.1 Utilisation de la liaison série (USB) avec l'Arduino

```
1 void setup() {  
  // Initialize the serial communication  
3  Serial.begin(9600);  
  }  
5  
6 void loop() {  
7  // Send the value of analog input 0  
  Serial.println(analogRead(A0));  
9  // Wait a bit for the analog-to-digital converter  
  // to stabilize after the last reading  
11 delay(10);  
  }
```

#### 3.2 Utilisation de la liaison série (USB) avec Processing

```
import processing.serial.*;  
2  
3 Serial myPort; // The serial port  
4  
5 void setup () {  
6  // List all the available serial ports.  
  println(Serial.list());  
8  // Open the first port listed.  
  myPort = new Serial(this, Serial.list()[0], 9600);  
10  // Don't generate a serialEvent()  
  // unless you get a newline character.  
12  myPort.bufferUntil('\n');  
  }  
14  
15 void serialEvent (Serial myPort) {  
16  // Get the ASCII string:  
  String inString = myPort.readStringUntil('\n');  
18  
19  if (inString != null) {  
20  // Trim off any whitespace:  
    inString = trim(inString);  
22  // Convert to an int and map to the screen height:  
    float inByte = float(inString);  
24  inByte = map(inByte, 0, 1023, 0, height);  
26  
    ...  
  }  
28 }
```

### 3.3 Gestion des entrées/sorties

La plupart des projets ont à manipuler les entrées et sorties numériques des Arduino. On peut par exemple faire un jeu de réflexe avec trois diodes électro-luminescentes et deux boutons poussoirs.

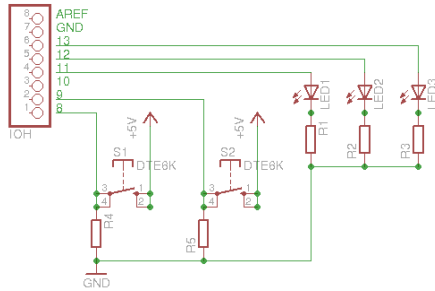


FIGURE 2 – Schémas de câblage

```
void setup() {
2  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
4  pinMode(11, OUTPUT);
  pinMode(9, INPUT);
6  pinMode(8, INPUT);
}

8
void loop() {
10  digitalWrite(11, LOW); // Set the LEDs off
  digitalWrite(12, LOW); // For player 2
12  digitalWrite(13, LOW); // For player 1
  int t = random(1,9); // Random waiting time
14  delay(t*1500);
  digitalWrite(11, HIGH); // Set the LED on

16
  int u1 = 0, u2 = 0; // Two players
18  while (!u1 && !u2) {
    u1=digitalRead(8);
20    u2=digitalRead(9);
  }
22  digitalWrite(11, LOW); // Set the LED off

24  if(u1 && !u2) { // Who is the first ?
    digitalWrite(13, HIGH); // set the LED on for player 1
26  }
  else if(!u1 && u2) {
28    digitalWrite(12, HIGH); // set the LED on for player 2
  }else {
30    digitalWrite(13, HIGH); // set the LED on
    digitalWrite(12, HIGH); // for both !
32  }
  delay(2000); // Wait for two seconds
34 }
```