

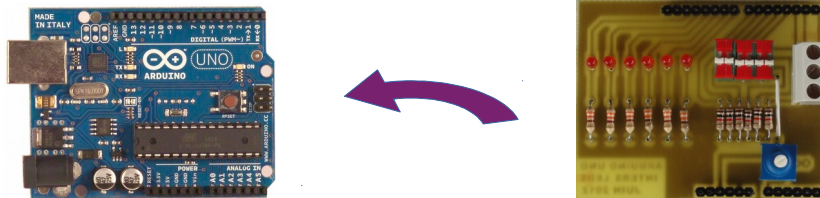
NOM :	CLASSE :
PRÉNOM :	

Condition :	<ul style="list-style-type: none"> <li>• Durée 2 heures</li> </ul>
Matériel :	<ul style="list-style-type: none"> <li>• un ordinateur sous Ubuntu avec les logiciels Arduino et Processing</li> <li>• une carte Arduino Uno, un shield interrupteurs et leds,</li> </ul>
Documents :	<ul style="list-style-type: none"> <li>• le sujet du TP</li> <li>• les cours sur Processing et Arduino</li> </ul>

L'interface graphique programmable Processing, véritable "couteau suisse logiciel", écrite en Java (multiplateforme donc...) par le MIT, libre et opensource, donne la possibilité de créer des interfaces graphiques côté PC pour communiquer avec le système Arduino. On peut ainsi commander à la souris ou au clavier la carte Arduino ou bien afficher sur le PC des graphiques à partir des données reçues depuis la carte Arduino. Processing permet également de réaliser de la capture vidéo, du traitement d'image, des applications réseau serveur ou client, de la lecture de son, de la reconnaissance vocale...



On utilise dans cette première partie la maquette Arduino Uno et la maquette shield interrupteurs et leds.



Pour rappel, cette carte shield est composée de 6 leds, 6 interrupteurs, un potentiomètre et un bornier connecté à la liaison série de l'Arduino. Le brochage de la carte Shield est le suivant :

broche	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	A0
nom	I2	I3	I4	I5	I6	I7	led8	led9	led10	led11	led12	led13	pot

I = Interrupteurs

pot = potentiomètre

## 1. Installer les bibliothèques sous Processing

Nous allons utiliser deux bibliothèques pour Processing.

- ✓ Pour installer cette bibliothèque, démarrez Processing,
- ✓ passez en mode Java. Cliquez sur *Outils* > *Ajouter un outil...* > *Libraries ...*
- ✓ Recherchez la bibliothèque *Arduino (Firmata)* puis installez-la.
- ✓ De la même façon recherchez puis installez la bibliothèque *controlP5*. Fermez la fenêtre.
- ✓ Si vous travaillez sous Linux, utilisez Nautilus, l'explorateur de fichiers, pour renommer le fichier `/sketchbook/libraries/arduino/library/Arduino.jar` en `arduino.jar` (remplacez le A majuscule par un a minuscule).
- ✓ Fermez Processing.



## 2. Télécharger le firmware dans l'Arduino



Pour que notre maquette Arduino Uno communique avec Processing, il nous faut installer le firmware StandardFirmata.

- ✓ Démarrez le logiciel Arduino. Ouvrez le fichier exemple Firmata > Standardfirmata. Téléchargez ce fichier dans l'Arduino. Fermer le logiciel Arduino.

**Votre Arduino est prêt à communiquer avec Processing.**

## 3. Commander les leds à partir de Processing

### 1.1 Allumer une led

L'objectif est d'allumer la led D13 à partir de Processing. On donne le programme sous Processing :

```
/* prog1_allumeD8.pde */  
  
import processing.serial.*;  
import cc.arduino.*;  
  
Arduino arduino;  
  
void setup()  
{  
  arduino = new Arduino(this, Arduino.list()[0], 57600);  
  arduino.pinMode(8, Arduino.OUTPUT);  
}  
  
void draw()  
{  
  arduino.digitalWrite(8, Arduino.HIGH);  
}
```

- ✓ Redémarrez Processing et copiez-collez ce programme sous le nom *prog1\_allumeD13.pde*. Testez.
- ✓ Retrouvez dans ce programme les éléments suivants :
  - mise à 1 de la sortie 8 de l'arduino
  - déclaration d'un objet arduino de type Arduino
  - déclaration de la broche 8 de la maquette arduino en sortie

### 1.2 Commander une led à partir du clavier

L'objectif est ici d'allumer la led D10 du shield lorsqu'on appuie sur la touche '1' du clavier et de l'éteindre lors de l'appui sur la touche '0'.

- ✓ Modifiez le programme précédent. Enregistrez le fichier sous *prog2\_ledetclavier.pde*. Testez.

Validation prof :

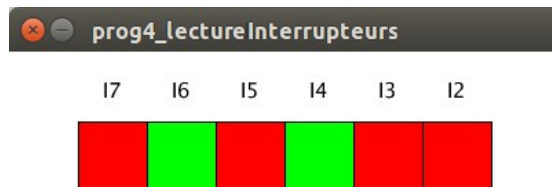
- ✓ Modifiez votre programme afin de commander la led D10 du shield avec une variation de la luminosité à partir des touches 0 à 9 du PC. L'éclairage varie en fonction du numéro de la touche appuyée : 0=led éteinte ; 9=led allumée pleine puissance. Enregistrez le fichier sous *prog3\_variationD10.pde*. Testez.

Validation prof :

### 1.3 Lire les états des interrupteurs

L'objectif est d'afficher dans la fenêtre de Processing les états des 6 interrupteurs de la maquette shield.

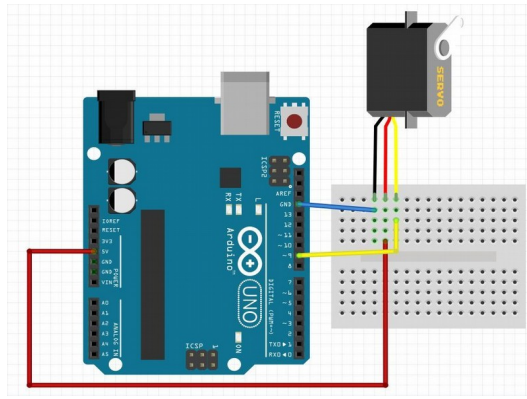
- ✓ Modifiez votre programme afin de réaliser cet affichage. La fenêtre sous Processing, de taille 400x150 pixels, ressemblera à celle ci-dessous : un carré vert pour un état haut, un carré rouge pour un état bas. Enregistrez le fichier sous *prog4\_lectureInterrupteurs.pde*. Testez.



Validation prof :

### 1.4 Faire bouger un servo moteur en fonction de la position de la souris

- ✓ Faire ce montage



- ✓ Dans les exemples, démarrer arduino (Firmata) > arduino\_servo.
- ✓ Modifier le programme pour utiliser la borne 9 de l'arduino
- ✓ Enregistrez le fichier sous *prog5\_servo.pde*. Testez.
- ✓ Compléter en français les commentaires du programme

Validation prof :

```

/*
arduino_servo
*/

import processing.serial.*;
import cc.arduino.*;

Arduino arduino;

void setup() {
  size(360, 200);

  // .....
  println(Arduino.list());

  // Modify this line, by changing the "0" to the index of the serial
  // port corresponding to your Arduino board (as it appears in the list
  // printed by the line above).
  arduino = new Arduino(this, Arduino.list()[0], 57600);

  // .....
  arduino.pinMode(4, Arduino.SERVO);
  arduino.pinMode(7, Arduino.SERVO);
}

void draw() {
  background(constrain(mouseX / 2, 0, 180));

  // .....
  // .....
  // .....
  arduino.servoWrite(4, constrain(mouseX / 2, 0, 180));
  arduino.servoWrite(7, constrain(180 - mouseX / 2, 0, 180));
}

```