

Info 2 SCP

Interaction Arduino & Processing

Damien Sauveron

<http://damien.sauveron.fr/>

Interactions Processing - Arduino

- Jusqu'à présent nous avons été capables d'écrire des programmes Processing pouvant traiter des structures de données avancés et/ou de manipuler des fichiers.
- Jusqu'à présent nous avons aussi été capables avec Arduino (Wiring) de mesurer des grandeurs physiques, d'enregistrer des données dans des fichiers ou d'agir sur l'environnement.
 - Nous avons aussi vu comment commander Arduino au travers d'un logiciel connecté sur le port série
- Nous allons voir comment écrire des programmes Processing interagissant avec Arduino

La bibliothèque `processing.serial`

- En préalable il faut importer `processing.serial`
- Avec `Serial.list()`, il est possible d'obtenir la liste des ports série disponibles sur votre ordinateur.
- Pour communiquer avec le périphérique brancher sur un port spécifique, il faut :
 - Déclarer une variable de type `Serial`
 - Créer la connexion en précisant celui qui crée le port (en général `this`), le nom du port et le débit voulu
 - Le débit doit être identique sur l'autre périphérique
 - Le `this` est nécessaire si on veut traiter les événements du port série (voir les interruptions sous Arduino – ici au travers du callback `serialEvent`)

Initialisation du port série

```
import processing.serial.*;

// Le port serie
Serial monPort;

void setup() {

    // Affiche tous les ports disponibles
    printArray(Serial.list());

    int num = askInteger("Selectionner le numero");

    // Ouvre le port au debit voulu (ici 9600 bauds)
    monPort = new Serial(this, Serial.list()[num], 9600);
}

void draw(){

    // Utilisation du port série
}
```

Les fonctions de la bibliothèque `processing.serial`

- `available()`
- `read()`
- `readChar()`
- `readBytes()`
- `readBytesUntil()`
- `readString()`
- `readStringUntil()`
- `buffer()`
- `bufferUntil()`
- `last()`
- `lastChar()`
- `write()`
- `clear()`
- `stop()`

Lecture de données

```
import processing.serial.*;

// Le port serie
Serial monPort;

void setup() {

    // Affiche tous les ports disponibles
    printArray(Serial.list());

    int num = askInteger("Selectionner le numero");

    // Ouvre le port au debit voulu (ici 9600 bauds)
    monPort = new Serial(this, Serial.list()[num], 9600);
}

void draw(){

    // Utilisation du port série
    while (monPort.available() > 0) {
        int inByte = monPort.read();
        println(inByte);
    }
}
```

Envoie de données

```
import processing.serial.*;

// Le port serie
Serial monPort;

void setup() {

    // Affiche tous les ports disponibles
    printArray(Serial.list());

    int num = askInteger("Selectionner le numero");

    // Ouvre le port au debit voulu (ici 9600 bauds)
    monPort = new Serial(this, Serial.list()[num], 9600);
}

void draw(){
}

void keyPressed() {
    switch (key) {
        case 'a': // touche 'a' pour allumé
            monPort.write("1");
            break;
        case 'e': // touche 'e' pour éteint
            monPort.write("0");
            break;

        case 'x': // x : pour terminé
            monPort.stop();
            exit(); // Quitter pour être sur que tout est ok
            break;
    }
}
```

Exemple 1 : Envoie et réception (coté Processing)

```
import processing.serial.*;

// Le port serie
Serial monPort;

void setup() {

    // Affiche tous les ports disponibles
    printArray(Serial.list());

    int num = askInteger("Selectionner le numero");

    // Ouvre le port au debit voulu (ici 9600 bauds)
    monPort = new Serial(this, Serial.list()[num], 9600);

}

void draw(){

    if (monPort.available() > 0) {
        // Conversion du int en char
        inChar = char(monPort.read());
        println(inChar);
    }

}
```

```
void keyPressed() {
    switch (key) {
        case ' ': // touche ' ' pour allumé
            monPort.write("1");
            break;
        default:
            monPort.write(key);
    }
}

void keyReleased() {
    switch (key) {
        case ' ': // touche ' ' pour éteint
            monPort.write("0");
            break;
    }
}
```

Exemple 1 : Envoie et réception (coté Arduino)

```
const int sortie_numerique = 13;

void setup() {
  // Vitesse de la communication = 9600 bauds
  Serial.begin(9600);
  pinMode(sortie_numerique, OUTPUT);
}

void loop() {
  char octet_recu;
  if (Serial.available() > 0) {
    // On lit ce que l'on reçoit via le port série
    octet_recu = Serial.read();

    if ((octet_recu == '1') || (octet_recu == '0')) {
      if (octet_recu == '1') {
        digitalWrite(sortie_numerique, HIGH);
      } else {
        digitalWrite(sortie_numerique, LOW);
      }
    }
    else
      Serial.print(octet_recu);
  }
  delay(10);
}
```