TP Communication entre Processing et l'Arduino

Comme vous l'avez vu dans le diaporama, nous allons faire communiquer le logiciel Processing avec l'Arduino.

Partie Arduino :

Tout d'abord lancer le programme Arduino '*Prog_Arduino'* que l'on vous a fournit. Vous pouvez le téléverser sur votre carte Arduino.
Brancher une palette de LED sur le port 22 et sur le GND (comme dans les autres TP).
C'est finit pour la partie Arduino.

Partie Processing :

Lancer le programme Processing que vous compléterez au fur et à mesure du tp :

Un programme Processing se divise en trois partie(comme le logiciel Arduino) une partie variable qui se trouve au début du programme

il se compose aussi d'une partie configuration :

```
void setup()
{

}
```

Et une partie similaire au *'Void loop()'* de l'Arduino qui ce répéteras tout le temps :

```
void draw()
{

}
```

Etape 1 :
Dans la partie SETUP vous pouvez changer la taille de la fenêtre grâce à la fonction *'size(Hauteur fenêtre, Largeur fenêtre);'* (nous vous conseillons une fenêtre de 600x400)

Puis nous pouvons mettre une couleur unis en fond :
La fonction est : *'background(r, v, b);'* Entre les parenthèse vous devez écrire a la place de *'r'* la quantité de rouge que vous voulez mettre, même chose pour le *'v',* pour le vert, et le *'b'* pour le bleu.

Par exemple :

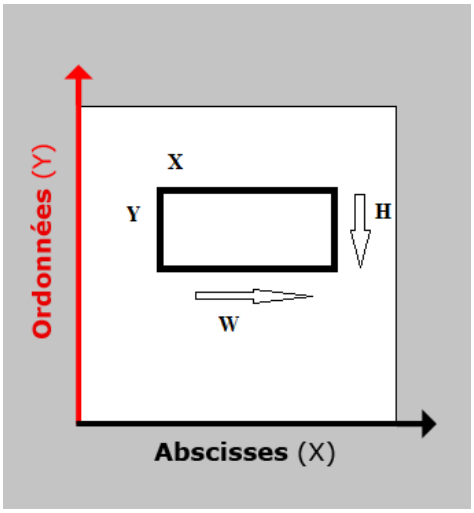
couleur blanche : *background(255 255, 255);*
couleur rouge :*background(255, 0, 0);*
couleur verte :*background(0, 255, 0);*
couleur bleu :*background(0, 0, 255);*

Etape 2 :

Maintenant on va voir comment afficher une image à l'écran.
Pour cela il va falloir une variable pour chaque image que nous utiliseront(ici on en utiliseront 2).
Une variable d'image ce déclare comme cela : *'PImage nom_de_la_variable ;'*
Nous allons déclarez deux variable image, une pour le bouton ON une pour le bouton OFF.
Donc on écriras : *'PImage imgON, imgOFF;'*

Maintenant nous allons remplir ces variables dans la partie *'setup'*. Il faut écrire le nom de la variable = a ce que l'on veut.
Ici on veut remplir la variable ON et OFF, nous allons donc écrire *'imgON = loadImage(« ON.png ») ;'* et *'imgOFF = loadImage("OFF.png");'*

On a maintenant charger les image dans les variables. Pour les afficher il faut écrire :
'image(variable_de_l'image, x, y, h, w) ;'
A la place de *variable_de_l'image* nous allons mettre nos *'imgON'* ou *'imgOFF'* .
Le *'x'* est l'emplacement en x de l'image sur l'écran.
Le *'y'* est l'emplacement en y de l'image sur l'écran.
Le *'h'* est la taille de l'image en hauteur a afficher sur l'écran.
Le *'w'* est la taille de l'image en largeur a afficher sur l'écran.



Nous allons afficher l'image ON (*'imgON'*) comme cela :
'image(imgON, 310, 50, 260, 260);'
et l'image OFF (*'imgOFF'*) comme cela :
'image(imgOFF, 50, 50, 260, 260);'

TP Communication entre Processing et l'Arduino

Nous allons afficher l'image ON seulement si la LED est éteinte.
Nous allons donc utiliser les conditions : *'if(condition)'* .
Pour savoir si la LED est allumée ou éteinte nous allons créer une variable binaire(1 ou 0) qui sera égale a *'true'* (1) ou *'false'* (0). Donc pour créer une variable binaire on utilise
'boolean variable = true/false'
Nous allons créer la variable ON qui seras de base égale a *'false'* de base
'boolean ON=false;'
Donc on va utiliser la conditions avec cette nouvelle variable :

```
'if(ON == true)
{
  //alors on affiche l'image OFF!
  image(imgOFF, 50, 50, 260, 260);
}
if(ON == false)
{
  //alors on affiche l'image ON!
  image(imgON, 310, 50, 260, 260);
}
```

Etape 3 :
Maintenant il va falloir faire que les images soient des boutons.
Pour les transformer il faut savoir si la souris est sur l'image et savoir si elle clique.
Pour cela nous allons utiliser les coordonnées de la souris qui sont *'mouseX'* et *'mouseY'* et l'appui sur un bouton de la souris qui est *'mousePressed'*.

Pour utiliser une condition avec plusieurs sous-conditions à l'intérieur nous utiliserons les connecteurs logiques *'et'* et le connecteur logique *'ou'* qui se traduisent par *'&&'* et *'||'* il faut les mettre après toutes les sous-conditions voulu tel que : *'if(condition1 == ?) && (condition2 == ? || condition2bis == ?)'*

Donc si *'mousePressed'* renvoie *'true'* que *'mouseX >= 310'* que *'mouseX <= 570'* et que *'mouseY >=50'* et enfin que *'mouseY <=310'* avec sa dans une condition on vérifiera si la souris clique sur l'image ON. Il faudra donc dire que *'ON=true'*

Maintenant si *'mousePressed'* renvoie *'true'* que *mouseX >= 50' que 'mouseX<=310'* et que *'mouseY >=50' et enfin que 'mouseY <=310'* avec ça dans une condition on vérifiera si la souris clique sur l'image *OFF. Il faudra donc dire que 'ON=false'*

Etape 4 :
On a donc nos boutons. Il faut maintenant terminer par dire si *'ON=true'* il faudra envoyer par USB la valeur 1 pour allumer la LED.
Et si *ON=false'* il faudra envoyer la valeur 0.

La fonction pour envoyer une valeur est *'myPort.write('VALEUR');'*.

TP Communication entre Processing et l'Arduino

Correction :

Voici le programme complet :

```
import processing.serial.*;
 Serial myPort;
boolean ON=false;
PImage imgON, imgOFF;

void setup()
{
  size(600, 400);
  println(Serial.list());
  myPort = new Serial(this, Serial.list()[2], 9600);
  myPort.bufferUntil('\n');
  imgON = loadImage("ON.png");
  imgOFF = loadImage("OFF.png");
  text("",165,195);
}

void draw()
{

  background(255);

  if (ON==false)
  {
    image(imgON, 310, 50, 260, 260);
  }
  else
  {
    image(imgOFF, 50, 50, 260, 260);
  }


  if ((mousePressed==true) && (mouseX>=310) && (mouseX<=570) && (mouseY>=50) && (mouseY<=310))
  {
    ON=true;
  }
  if ((mousePressed==true) && (mouseX>=50) && (mouseX<=310) && (mouseY>=50) && (mouseY<=310))
  {
    ON=false;
  }

  if (ON == true)
    {
      myPort.write('1');
    }
    else
    {
      myPort.write('0');
    }
}
```

Voila nous avons créer un programme qui permet allumer une LED grâce à l'appui sur des boutons.