

# IHM côté PC

Pour pouvoir piloter le banc à distance, il faut construire une IHM minimaliste côté PC qui fait également tourner le logiciel d'acquisition du Nitron.

L'idée qui vient naturellement est de faire cela en Processing, puisqu'on a déjà l'Arduino programmé dans un dialecte voisin.

Le point le plus "sophistiqué" (qu'il faut également résoudre en premier) est de proposer une liste ou un menu déroulant permettant de choisir le port USB sur lequel le M5 est connecté.

Une recherche Google rapide fait remonter une bibliothèque : [ControlP5](#) dont [les exemples](#) sont prometteurs. Hélas, la bibliothèque est ancienne, et les exemples ne marchent pas tels quels. La syntaxe est globalement hermétique.

Une alternative est la bibliothèque G4P, qui semble encore plus complexe.

Je tombe enfin sur un [programme](#) qui permet de tracer boutons et listes de choix, sans recours à une bibliothèque : c'est moche, mais c'est intelligible !



Il faut créer déclarer les objets boutons :

```
// other buttons
Button btn_serial_S;
Button btn_serial_IZ;
```

Puis les instancier dans le `setup()` :

```
void setup() {
  // set the window size
  size (640, 480);

  // create the buttons
  ...
  btn_serial_S = new Button("S", 20,200, 100, 25);
  btn_serial_IZ = new Button("IZ", 20,230, 100, 25);
```

Puis définir leur comportement dans la fonction `mousePressed()` :

```

// S button clicked
if (serial_port != null) {
    if (btn_serial_S.MouseIsOver()) {
        serial_port.write("S\n");
        println("message S");
    }
}
// IZ button clicked
if (serial_port != null) {
    if (btn_serial_IZ.MouseIsOver()) {
        serial_port.write("IZ\n");
        println("message IZ");
    }
}
}

```

Enfin, il faut les dessiner dans la boucle principale `draw()` :

```

btn_serial_S.Draw();
btn_serial_IZ.Draw();

```

C'est donc également plutôt laborieux, puisqu'il faut intervenir à de nombreux endroits du code pour chaque bouton.

Voici le [code complet IHM\\_XRF\\_v2.pde](#) : la communication avec le M5 fonctionne.

Même si l'IDE est plutôt prévenant et pointe les erreurs assez efficacement, je me pose la question d'utiliser un autre langage avec un framework un peu récent. Par exemple python avec TkInter, ou PyQt ou d'autres. [Une review des GUI python ici](#).

L'avantage, c'est qu'il existe des builder comme PyQtDesigner... Mais est-ce nécessaire vu la simplicité ici ?

---

Revision #7

Created 8 February 2025 06:32:08 by Christian Simon

Updated 15 February 2025 10:37:10 by Christian Simon