

Documentation des codes utilisés dans les ateliers 2 et 3

Comment est construite cette documentation ?

Cette documentation est construite autour de deux entités : le code qui est commenté et ce document qui illustre le principe de fonctionnement, les branchements et les algorithmes mis en place.

Principe général des codes

L'Arduino récupère l'état logique (1 ou 0) des fourches optiques et envoie l'information ainsi que le temps auquel cette information a été prise vers l'ordinateur via une connexion série. L'ordinateur gère ensuite le traitement.

Fonctionnement

Le fichier Arduino correspondant à l'atelier 2 doit être envoyé sur la carte Arduino via ArduinoIDE. Le fichier .py est le fichier source qu'il faudra modifier pour pouvoir changer la façon dont on gère la connexion entre l'Arduino et l'ordinateur. Le fichier .exe en lien avec le dossier auquel il est attaché est le fichier à lancer sur l'ordinateur pour commencer la liaison série. Le fichier .exe est un fichier exécutable Windows qui a été créé grâce à cx_freeze à partir du code Python.

Atelier 2



eruches_app_atelier_2.ino

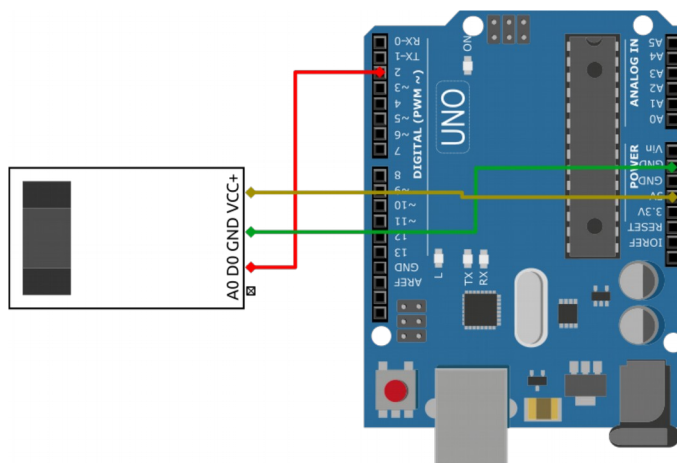


eruches_atelier2_compteur.py



eruches_atelier2_compteur.exe

Branchements de la carte Arduino



Brancher la carte Arduino à l'ordinateur. Lancer l'exécutable .exe.

Algorithme de traitement

```
compteur ← 0
x_nouveau ← valeur
TANT QUE Vrai
```

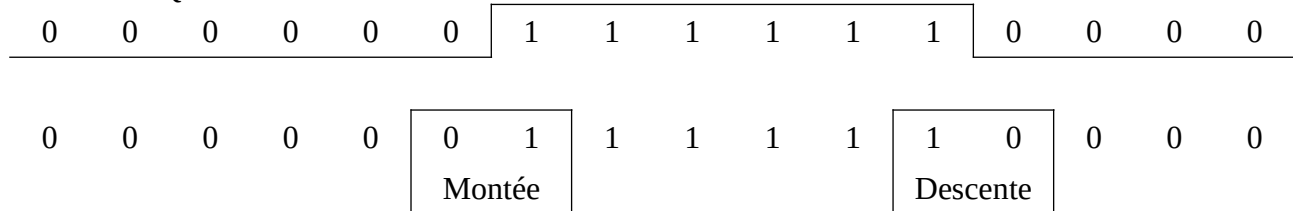


```

x_vieux ← x_nouveau
x_nouveau ← valeur
SI x_nouveau différent de x_vieux
    compteur+=x_nouveau
FIN SI

```

FIN TANT QUE



On détecte à la fois les montées et les descentes mais en analysant bien l'algorithme on se rend compte que la descente n'a aucune influence sur le compteur. On s'abstiendra donc de présenter l'algorithme sous cette forme en classe.

Nous avons fait le choix de présenter l'algorithme sous la forme suivante :

SI la valeur précédente est un 0 et la valeur courante un 1 ALORS on passe d'un état logique bas à un état logique haut donc on **incrémente le compteur d'1**.

On pensera à adapter la présentation de l'algorithme au niveau de la classe.

Atelier 3



eruches_app_atelier_3.ino

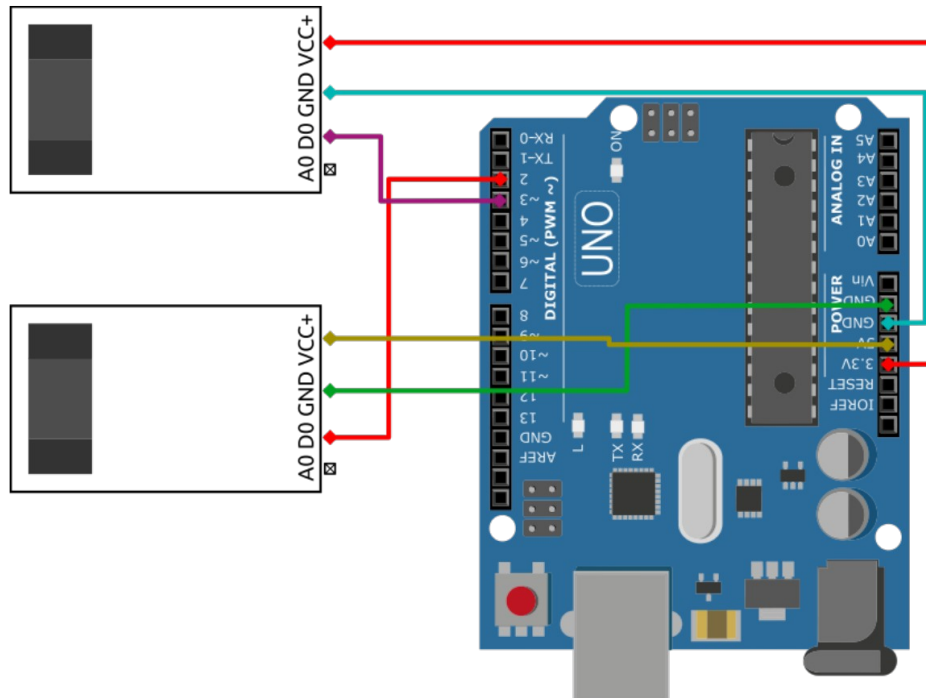


eruches_atelier3_compteur.py



eruches_atelier2_compteur.exe

Branchement de l'arduino et des fourches optiques



Brancher la carte Arduino à l'ordinateur et lancer l'exécutable .exe

Algorithme de traitement

On choisi comme convention que le capteur1 est à l'extérieur et le capteur2 est à l'intérieur de la ruche

compteur ← 0

#Voir le choix de structure de donnée choisi dans Python (liste)

x_nouveau1 ← valeur1

x_nouveau2 ← valeur2

TANT QUE Vrai

 x_vieux1 ← x_nouveau1

 x_vieux2 ← x_nouveau2

 x_nouveau1 ← valeur1

 x_nouveau2 ← valeur2

 SI (x_vieux1 == 0) ET (x_nouveau1 == 1) ET (x_vieux2 == 1) ET (x_nouveau2 == 1)

 ALORS compteur -=1

 SINONSI (x_vieux1 == 1)ET(x_nouveau1 == 1)ET(x_vieux2 == 0)ET(x_nouveau2 == 1)

 ALORS compteur +=1



FIN SI FIN TANT QUE

Cas 1

Voie1

0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Voie2

Voie1

0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0

Voie2

compteur +=1

Cas 2

Voie1

0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Voie2

Voie1

0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0

Voie2

compteur -=1

La présentation de ces schémas fonctionnait plutôt bien en classe. Pourquoi pas utiliser des couleurs à la place des 0 et des 1.



Suggestions et idées d'amélioration

- Modifier les codes existants afin que les programmes fonctionnent avec un programme unique dans l'Arduino et non deux comme actuellement.