

Encodeur Rotatif

D.Moreaux

19 avril 2021

1 Principe

Les encodeurs rotatifs (comme ceux qu'on trouvait dans les anciennes souris) utilisent deux interrupteurs (mécaniques ou optiques) qui se déclenchent avec un décalage de $1/4$.

Ainsi, dans un sens de rotation, on a la succession

S1 haut	S2 haut	S1 bas	S2 bas	S1 haut	S2 haut	...
---------	---------	--------	--------	---------	---------	-----

Dans l'autre sens de rotation, la séquence est inversée.

On peut détecter le déplacement en détectant les changements d'un des signaux (par exemple S1) et en regardant la valeur de l'autre signal au moment du changement. Selon le sens de rotation, l'autre signal sera soit identique, soit opposé.

La lecture peut se faire avec 3 résolutions différentes :

- 1x** Un signal sert d'horloge. Lors d'une transition bas vers haut, on lit l'autre signal pour déterminer s'il faut augmenter ou diminuer la valeur
- 2x** Même traitement que pour le 1x mais en détectant également les transitions haut vers bas
- 4x** Lorsqu'un des deux signaux change, on vérifie la valeur de l'autre signal pour déterminer le sens.

La bibliothèque Encoder travaille en mode 4x.

Les Arduinos peuvent configurer certaines broches pour qu'elles déclenchent une interruption quand la valeur de la broche change. Utiliser une ou les deux broches en interruption permettra d'obtenir une meilleure résolution (et moins d'erreurs).

Sur un Arduino *standard*, les broches en question sont les broches 2 et 3. Si on n'utilise pas de broche avec interruption, il faut appeler manuellement la lecture de l'encodeur de façon répétée¹.

2 utilisation

Une fois la bibliothèque `Encoder` installée, on pourra l'utiliser

```
#include <Encoder.h>
```

```
Encoder bouton(2,3);
```

Par la suite, il suffira de lire la valeur de `bouton.read()` (long) pour connaître la position.

La fonction `bouton.write(val)` permet de modifier la valeur actuelle de l'encodeur (par exemple, la remettre à 0 ou forcer des limites).

3 Bornes avec interruptions

Pour l'Arduino UNO, l'Arduino Nano, le RGBDuino, ... (ATMega328), les bornes 2 et 3 proposent des interruptions.

Pour l'Arduino Leonardo, l'Arduino Micro Pro, ... (ATmega32U4), les bornes 0, 1, 2 et 3 proposent des interruptions.

Pour l'Arduino Mega, ... (ATmega2560), ce sont les bornes 2, 3, 18, 19, 20 et 21 qui proposent des interruptions.

Les Arduino basés sur un processeur ARM proposeront généralement des interruptions sur toutes les bornes

1. si le délai entre deux appels est trop long, on risque de perdre des transitions