

DualEncoders

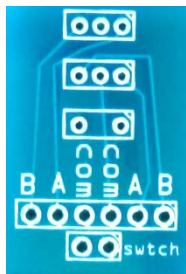
TUTORIEL

QUEST-CE QUE DualEncoders?

DualEncoders est une bibliothèque qui vous permet d'utiliser aisément des encodeurs concentriques qui ont deux pas par clic.

COMMENT CELA FONCTIONNE?

Chaque encodeur possède trois broches nommées A, B et Com. Les broches A et B sont reliées à des broches digitales de l'Arduino. La broche Com est reliée à la terre (broche Gnd de l'Arduino). Une des broches nommées « switch » est reliée à une broche digitale de l'Arduino et l'autre à la terre.



```
//=====
// Encodeur      Arduino

// externeB-----3
// externeA-----2
// Com-----GND--+
// Com-----+
// interneA-----4 |
// interneB-----5 |
// switch-----6 |
// switch-----+
//-----
```

Afin de pouvoir se servir de la bibliothèque, au début du sketch, on doit l'inclure à l'aide de la ligne suivante :

```
#include <DualEncoders.h>
```

Nous pouvons maintenant créer un objet DualEncoder. Dans l'exemple suivant, l'objet portera le nom « radioCOM1 ». Entre parenthèses, **dans l'ordre**, apparaissent les numéros de broche suivants : (externeA, externeB, interneA, interneB, switch). Nous pouvons créer autant d'objets DualEncoders que désiré. Il suffit de leur donner des noms différents.

```
DualEncoders radioCOM1(2, 3, 4, 5, 6);
```

Dans la section setup() du sketch, nous allons ajouter la ligne suivante, qui initialise pour nous les 4 broches des encodeurs et celle de l'interrupteur (switch). « pinMode(x, INPUT_PULLUP) »

```
radioCOM1.begin();
```

Les encodeurs sont différenciés par les noms « inner » et « outer ». Un appel à ces deux méthodes permet de connaître la direction dans laquelle les encodeurs sont tournés : CW (dans le sens horaire) et CCW (dans le sens antihoraire). Si l'encodeur n'a pas été tourné ou qu'il n'est pas au bout de sa course (clic), ces méthodes retourneront 0 (zéro).

```
int externe = anEncoder.outer();1
int interne = anEncoder.inner();
```

La lecture de l'interrupteur se fait avec la ligne suivante. Quand l'Interrupteur est pressé, la méthode attend qu'il soit relâché et retourne 0 (zéro). Sinon, elle retourne 1 (un).

```
byte interrupteur = radioCOM1.readSwitch();
```

¹ On doit utiliser une variable de type int. (CW=1 et CCW=-1)

EXEMPLE

```
#include <DualEncoder.h>

DualEncoder radioCOM1(2, 3, 4, 5, 6);

//setup=====
void setup() {
    Serial.begin(9600);           //Pour utiliser le moniteur série
    radioCOM1.begin();           //Pour initialiser les encodeurs
    Serial.println("Pret");       //Avertir l'utilisateur
} //setup-----

//loop=====
void loop(){
    int stat;
    stat = radioCOM1.readInner();
    if (stat == CWW) Serial.println("sim/radios/stby_com1_coarse_down");
    if (stat == CW) Serial.println("sim/radios/stby_com1_coarse_up");
    stat = radioCOM1.readOuter();
    if (stat == CCW) Serial.println("sim/radios/stby_com1_fine_down");
    if (stat == CW) Serial.println("sim/radios/stby_com1_fine_up");
    if (radioCOM1.readSwitch() == 0) Serial.println("sim/radios/com1_standby_flip");
} //loop-----
```

Ceci est la version 1.1 : Un détecteur de rebonds et de parasites électromagnétiques ultrarapide a été ajouté. Il en résulte des lectures beaucoup plus stables.

Le code a été complètement réécrit. Cette version est rétro-compatible avec la version 1.0. Seule la méthode .begin() a été ajoutée afin d'initialiser les broches dans la section setup(). Dans la version 1.0, l'initialisation se faisait lors de l'instanciation de l'objet. J'ai été avisé qu'il était plus avisé de communiquer avec la partie matérielle dans la section setup().

J'espère sincèrement que cette bibliothèque vous sera utile dans vos projets.

Jacques Bellavance