

PONGUINO Séance 2 08/01/2020

1/ Pendant les vacances

Pendant les vacances, j'ai relié le moteur à l'arduino grâce à un module L298. Grâce à une alimentation, nous pouvons désormais contrôler le moteur en 12V et avec l'Arduino (alimenté en 5V bien sûr). J'ai également fait des recherches concernant les encodeurs pour pouvoir les utiliser lors de cette séance.

Lors de cette séance :

2/ Problèmes tiges en acier

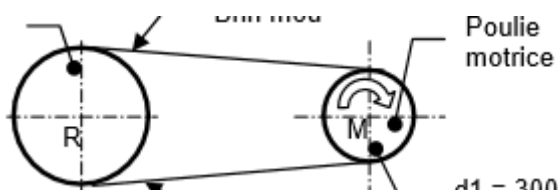
Tout d'abord, les tiges en acier que nous avons commandé n'ont pas le bon diamètre (le module ne glisse pas dessus), je suis donc allé en acheter moi-même après la séance (budget perso, moins de 10€) pour être sûr que le module s'adapte correctement dessus.

3/ Utilisation encodeurs moteurs

J'ai ensuite cherché sur internet un code arduino pour les encodeurs des moteurs que nous avons à disposition (je l'avais trouvé pendant les vacances mais ne l'avait pas encore appliqué). Les encodeurs nous renvoient en tant réel la vitesse de rotation du moteur. Cela nous permet de connaître la vitesse de notre courroie avec la formule $v = \omega R$.

VITESSE LINEAIRE D'UNE COURROIE : V

$V = \omega \times R$	avec	$\omega = \frac{2\pi n}{60}$	(ω est la vitesse angulaire)
m/s rad/s m		rad/s tr/min	



Ainsi, nous pourrions savoir la position en temps réel du module sur la tige en fer (cela nous permettra donc de connaître la position de la balle et des raquettes).

Voici le code pour connaître la vitesse de rotation du moteur (en tr/min) :

```
encoder$
#include <TimerOne.h> // Pour le timer
const int ENA=3;
const int IN1=4;
const int IN2=5;

const int ENCODEURA=7;
const int ENCODEURB=2;

volatile int count =0; // comptage de tick d'encodeur qui sera incrémenté sur interruption " On change " sur l'interruption matériel 0 qui se fait sur le pin 2
volatile double speed =0; // vitesse du moteur
volatile byte laststate =0; // état précédent de l'encodeur

void setup() {

  pinMode(ENA,OUTPUT);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);

  pinMode(ENCODEURA, INPUT_PULLUP);
  pinMode(ENCODEURB, INPUT_PULLUP);

  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);

  attachInterrupt(0,counter, CHANGE); // on crée l'interruption sur changement sur la pin 2 => interruption 0, la routine counter va se faire toute seule sans l'appeler à chaque changement

  Timer1.initialize(100000); // On définit le timer : 100000 microsecondes ( 0.1 sec - or 10Hz )
  Timer1.attachInterrupt( timerIsr ); // attach the service routine here la fonction timerIsr est une routine qui va se faire automatiquement 10* par secondes sans l'appeler

  Serial.begin(9600); // Définit vitesse de transmission série

}

void loop() { //le code dans la partie loop n'est pas important, il a simplement permis à faire des premiers test
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(ENA,HIGH);
  Serial.println(speed);
  delay(1500);
  Serial.println("arrêt");
  digitalWrite(ENA,LOW);
  delay(30000);
}

// Encoder counter 0

void counter()
{
  byte state=PIN0;

  state=~B01111011; // mask pour ne regarder que les changements sur 2 et 4
  // Modifier le MASK B01111011 par BXXXXXXX mettre des 0 là où sont les pins utilisés par l'encodeur
  if( state!=laststate)
  {
    (((state&(1<<ENCODEURA))>>ENCODEURA)^(state&(1<<ENCODEURB))>>ENCODEURB)?count--:count++;
    laststate=state;
  }
}

// Timer event to calculate speed from counters

void timerIsr()
{
  speed=count; // on a le nombre de tick ( signé ) par unité de temps = vitesse
  count=0; // On remet le compteur de tick à 0
}
```

Nous avons testé à différentes vitesses du moteur, et les vitesses en tr/min renvoyées étaient cohérentes.

4/ Microrupteurs

Avec William, nous avons remarqué que nous aurons de besoin de microrupteurs (bouton de butée). Tout d'abord, lorsque nous allumerons notre jeu, nous ne connaissons ni la position des raquettes, ni la position de la balle. Comment faire en sorte de connaître cette position ?

Exemple pour une raquette (même principe pour la balle) :

Nous fixerons un microrupteurs à la raquette, du côté gauche par exemple. Lors du lancement de notre jeu, nous ferons déplacer le moteur qui déplace la raquette à une faible vitesse vers la gauche jusqu'à ce bouton de butée touche un obstacle de fin de course et s'active. Ainsi, le moteur s'arrêtera, et nous pourrons initialiser cette position de la raquette comme « 0 » par exemple.

De plus les microrupteurs nous permettront de « réinitialiser » la position de la base lorsque la balle touchera un coté du plateau. En effet, si la position de la balle n'est connu que grâce à la vitesse de rotation des moteurs, la précision de l'emplacement de la balle sera de moins en moins précise.