GAROFALO Clément séance 3 10/01/2020

Ponguino

1/ Déplacement raquette avec potentiomètre

Nous nous sommes rendu compte que nous ne pourrons pas assurer le déplacement de la raquette avec le potentiomètre (ou que cela allait être compliqué alors qu'il existe d'autres solutions plus simples, voir plus bas). En effet, si l'on tourne trop rapidement le potentiomètre, nous faisons face à un problème : par exemple, on peut se retrouver avec une raquette au milieu de notre jeu alors que la position du potentiomètre est à une extrémité (par exemple, si la résistance du potentiomètre correspond à 0 Ohm, notre raquette doit se situer sur une extrémité, or régulièrement ce n'est pas le cas, donc certaines positions de la raquette devienne inaccessible)).

Nous avons donc décider d'utiliser un encodeur rotatif.

2/ Réflexion moteur

Nous avons trouvé une référence de moteur CC qui nous conviendrait pour le déplacement de la balle (il en faudrait 2 exemplaires). Mais nous voudrions aussi tester des moteurs pas à pas, qui seraient très avantageux pour nous car très précis (donc la position de la balle serait précise). Cependant ces moteurs ont en général une vitesse de rotation trop faible pour notre projet, ou alors un couple trop faible. Lors de la prochaine séance, nous verrons si l'on peut mettre 2 moteurs pas à pas sur une seule courroie afin d'augmenter le couple total, et si cela est plus simple qu'un seul moteur CC (avec les moteurs pas à pas à disposition).

3/ Déplacement raquette avec encodeur rotatif

Nous allons maintenant assurer le déplacement des raquettes avec un encodeur rotatif. Lorsque l'on tournera dans un sens l'encodeur, selon ce sens, la raquette se déplacera à gauche ou à droite.

Nous allons également utiliser 2 microrupteurs afin de savoir lorsque la raquette a atteint une extrémité (et ainsi éviter que le moteur continuer de tourner alors que la raquette fait face à un obstacle). Dans le futur, ces microrupteurs permettront aussi de réinitialiser la position de la raquette (car nous aurons besoin de connaitre la position de la raquette, surement de cette façon : elle sera calculée selon la vitesse et le sens des rotations des moteurs, puis ajusté avec un émetteur/récepteur).

J'ai eu du mal à coder le déplacement de la raquette avec l'encodeur. En effet, un delay dans mon programme perturbait le bon fonctionnement de l'encodeur, qui renvoyait des valeurs incohérentes, et j'ai pris du temps avant de me rendre compte que le problème venait de là. J'ai donc utilisé la fonction millis() plutôt que delay() afin de ne pas interrompre le programme et de ne plus avoir ce problème).

Voici le code : (la partie avec les microrupteurs n'est pas encore faite)

deplacement_raquette_encodeurs_Vsimple §

```
//moteur
const int ENA=9;
const int IN1=10;
const int IN2=11;

//encodeurs
int PinCLK=2; //CLK
int PinDT=3;    //DT
int PinSW = 4;//SW

static long val = -1;    // Au 1er démarrage, il passera à 0. valeur de l'encodeur
int PinCLKLast = LOW;   // variable qui permet de savoir lorsqu'on tourne l'encodeur
int n = LOW;   // variable qui permet de savoir lorsqu'on tourne l'encodeur
int ancien_val=0;
int val_mot=0;  //valeur du moteur (si en marche ou non)

long ancien_millis=0; //

void setup() {
  //encodeur
  pinMode (PinCLK,INPUT);
  pinMode (PinDT,INPUT);
  pinMode (PinSW,INPUT);
  Serial.begin (9600);

  //moteur
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(ENA,OUTPUT);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);

  delay(200);//pour regler pb ancien_millis
}
```

```
void loop ()  {
  readVal();

if (val!=ancien_val){ //cas où on va mettre en marche le moteur
  ancien_millis=millis();
  if (val<ancien_val){
    digitalWrite(IN2,LOW);
    digitalWrite(IN1,HIGH);}

  else if (ancien_val<val){
    digitalWrite(IN2,HIGH);
    digitalWrite(IN1,LOW);}

  val_mot=1;
}
else{
  val_mot=0;
}

if ((val_mot==1)|| (millis()-100<ancien_millis)){
  digitalWrite(ENA,HIGH);
  //Serial.println("moteur en marche");
  }

else{
  digitalWrite(ENA,LOW);
  //Serial.println("moteur éteint");
  }

  ancien_val=val;
 }


void readVal(){ //attribut à la variable val la valeur de l'encodeur
/* ce code en commentaire sera utilisé si on a besoin de récupérer la valeur du bouton de l'encodeur
   if (!(digitalRead(PinSW))) {      // Reset la position si on appui sur le potentiomètre
     encoderPos = 0;
     //Serial.println("Reset position");
   }
*/
   n = digitalRead(PinCLK);

   if ((PinCLKLast == LOW) && (n == HIGH)) {

     if (digitalRead(PinDT) == LOW) {
       val--;

     } else {
       val++;
       }

     Serial.println(val);
   }
   PinCLKLast = n;

 }
```

Voici une vidéo du résultat : https://www.youtube.com/watch?v=YPzzbAusFnI&feature=youtu.be