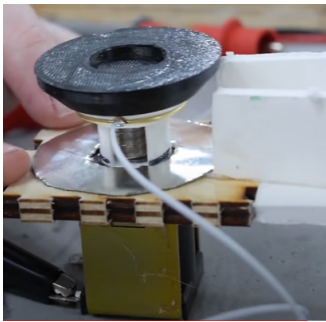


Compte rendu séance 4 Arduino

1) Bumpers:

Avant le début de la séance, nous avons adapté les dimensions des pièces utilisées pour les bumpers et les avons imprimées. Nous avons également imprimé un support pour fixer le solénoïde en dessous de la planche. Pendant la séance, nous avons testé le bon fonctionnement du mécanisme ainsi que l'activation du solénoïde (modèle 0630B 0.4-8N 10 mm) alimenté en 24V.

En ce qui concerne le déclenchement des bumpers, j'envisage de placer un anneau métallique sur la partie supérieure du bumper. Nous avons également discuté avec l'un des professeurs qui nous a suggéré de remplacer le ruban de cuivre par une plaque métallique, comme indiqué dans la photo ci-dessous en exemple :



Exemple de cette méthode (crédit : element14 presents)

La méthode proposée par le professeur consiste à créer un trou à l'aide d'une graveuse laser de manière à ce que la plaque métallique soit à ras de la planche et n'influence pas la trajectoire de la bille. Nous avons réalisé un premier prototype des bumpers avec la plaque métallique placée sur la planche.



Comme vous pouvez le constater, il nous manque encore l'anneau métallique à fabriquer. Nous envisageons d'ajouter un passage pour le câble de l'anneau métallique dans la pièce supérieure des bumpers et d'ajuster la hauteur de cette pièce afin d'obtenir plus de puissance lorsque la balle entre en contact avec les bumpers.

Photo d'un bumper avec une plaque métallique

J'ai testé le fonctionnement du dispositif détecteur de contact avec deux morceaux de feuille d'aluminium en utilisant le code suivant :

```

int s=5;
int res;
void setup() {
  Serial.begin(9600);
  pinMode(5,INPUT);
  pinMode(4,OUTPUT);
  digitalWrite(4,HIGH);

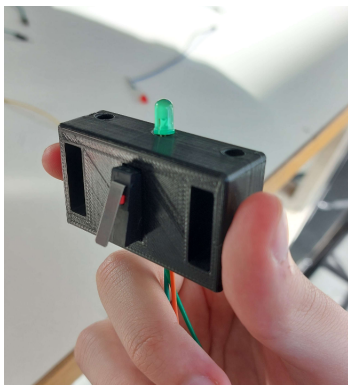
  // put your setup code here, to run once:
}

void loop() {
  res=digitalRead(5);
  if(res==1){
    Serial.println("yes");
  }
  else{
    Serial.println("no");
  }
}

```

2) Cibles:

Lors des séances précédentes, nous avons fabriqué la cage imprimée en 3D pour les cibles. Durant cette séance, j'ai soudé les fils et les résistances sur les leds. J'ai remarqué que les microswitches ne peuvent pas être installés correctement lorsque les leds et leurs fils sont dans la cage, donc nous devons modifier la profondeur de la cage pour avoir suffisamment d'espace pour les leds et les microswitches.



Exemple de la position du microswitch lorsque la led est installée.

J'ai retravaillé le code de test des microswitches pour éviter l'enregistrement de plusieurs entrées lorsque le bouton est appuyé seulement une fois en raison du phénomène de rebond d'un interrupteur. Et cela en utilisant la bibliothèque ezButton.

```

#include <ezButton.h>

ezButton limitSwitch(7); // create ezButton object that attach to pin 7;
int score;

void setup() {
  Serial.begin(9600);
  limitSwitch.setDebounceTime(10); // set debounce time to 10 milliseconds
}

void loop() {
  limitSwitch.loop(); // MUST call the loop() function first

  if(limitSwitch.isPressed())
    score++;
    Serial.println(score);

  if(limitSwitch.isReleased())
    Serial.println(score);

  int state = limitSwitch.getState();
  if(state == HIGH)
    Serial.println("microswitch: non appuyé");
  else
    Serial.println("microswitch: appuyé");
}

```