

# Faire de la physique avec ARDUINO – 9

Laurent FAURE, Nicolas HERVE, Gilles ESPINASSE (ENSFEA)



Dans ce neuvième article nous allons créer un projet Arduino qui va nous permettre de piloter une maquette reproduisant le fonctionnement d'un relevage de tracteur. Nous allons ajouter étape par étape les différents capteurs et actionneurs. Les programmes n'étant pas construits par des informaticiens, ils peuvent être largement améliorés. De même, la maquette pourra être modifiée en fonction de vos besoins.

## L'attelage des machines agricoles portées : le système de relevage trois points

### 1. Principe de fonctionnement

Lorsque l'on souhaite atteler un outil porté à un tracteur agricole, on utilise un attelage trois points. Il est constitué de bras de relevage, de bras inférieurs, de stabilisateurs latéraux réglables, de chandelles réglables et enfin d'une barre de poussée réglable. Cet ensemble est actionné par l'hydraulique du tracteur et permet d'accoupler une machine portée, c'est-à-dire une machine dont l'ensemble du poids est supporté par le tracteur lors des phases de transport ou de manœuvre. Lors des différents travaux, on pourra notamment réaliser différents réglages au niveau de l'attelage permettant d'utiliser la machine en sécurité et d'obtenir le travail souhaité. Sur les tracteurs actuels, la plupart des attelages trois points sont équipés de différents systèmes automatisés permettant un contrôle électrohydraulique de celui-ci.



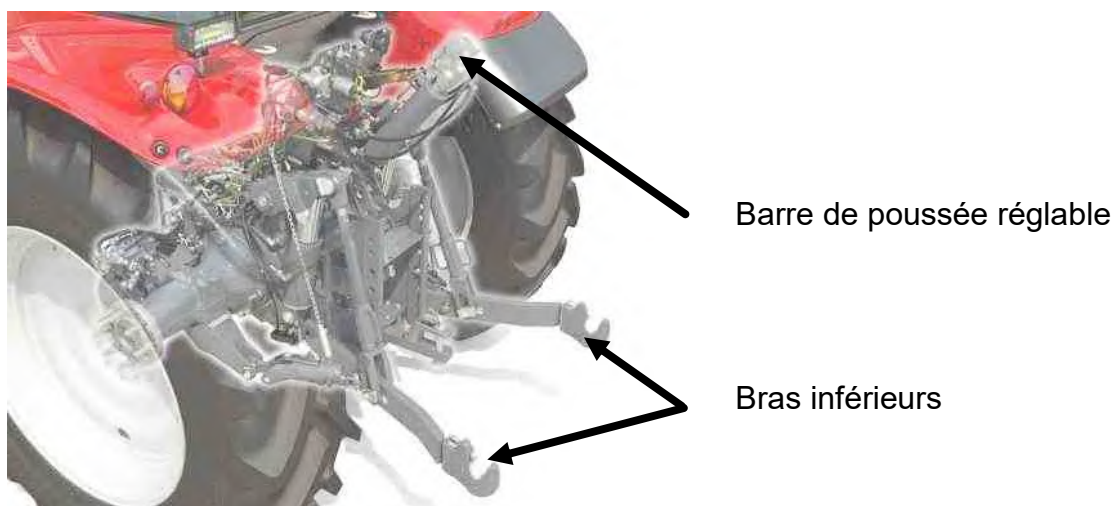
*Vue d'un tracteur et d'une charrue portée*

Parmi les fonctions électroniques d'un attelage trois points, on trouve un contrôle de la position, un contrôle de la vitesse de descente, un réglage du contrôle d'effort et une position haute de l'outil pour les manœuvres.

Le contrôle de position permet de régler de manière précise la hauteur des bras inférieurs de relevage. Le contrôle de la vitesse permet d'ajuster la descente de manière à éviter tout choc brusque avec le sol, il s'ajuste en fonction du poids de chaque outil et du travail que l'on est en train de réaliser. Le contrôle d'effort permet lors de travaux lourds d'avoir une

vitesse d'avancement de l'ensemble tracteur-outil constante en ajustant la profondeur de travail de l'outil. Enfin, la position haute permet, à l'aide d'un potentiomètre de positionner l'outil directement hors sol afin de pouvoir faire les manœuvres nécessaires en sécurité avant de reprendre le travail avec les réglages précédents.

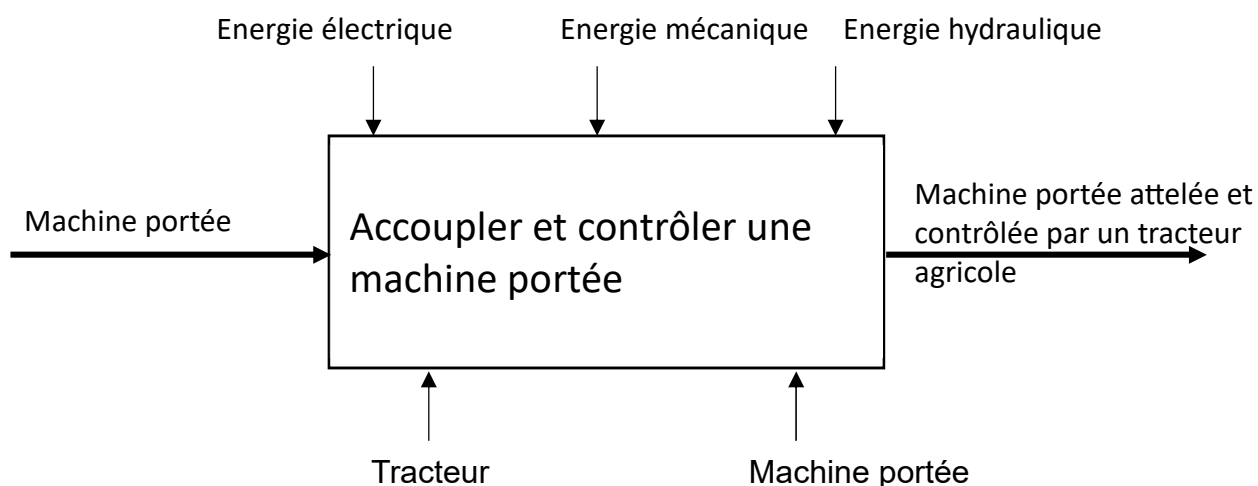
Nous avons repris ici quelques-unes des fonctions d'un attelage trois points à asservissement électrohydraulique d'un tracteur agricole qui ont été mobilisés sur la maquette pédagogique.



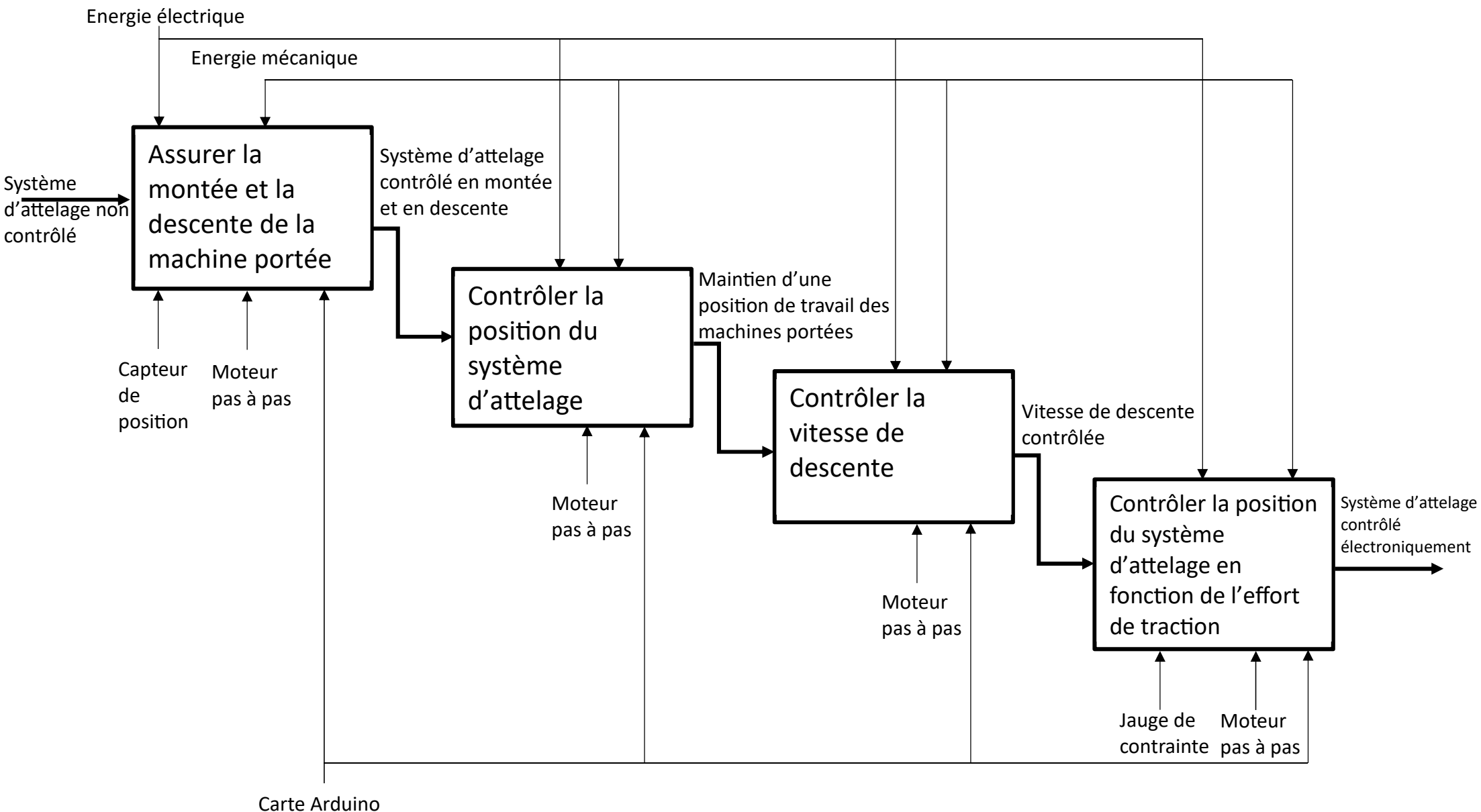
*Vue d'un attelage trois points arrière de tracteur agricole*

## 2. Analyse fonctionnelle

L'analyse fonctionnelle proposée ici présente la fonction globale d'un système d'attelage trois points du tracteur agricole, ainsi que les fonctions reprises dans la réalisation de la maquette, et non pas les nombreuses autres fonctions présentes sur les tracteurs actuels.



A0	Système de relevage trois points
----	----------------------------------

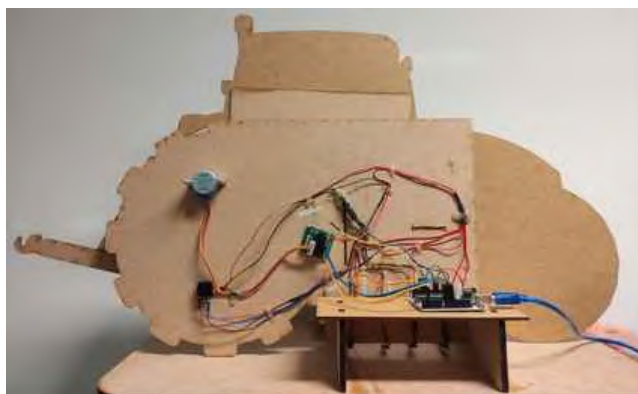


A2 Contrôle électronique du système d'attelage trois points

### 3. Présentation de la maquette



Vue de face

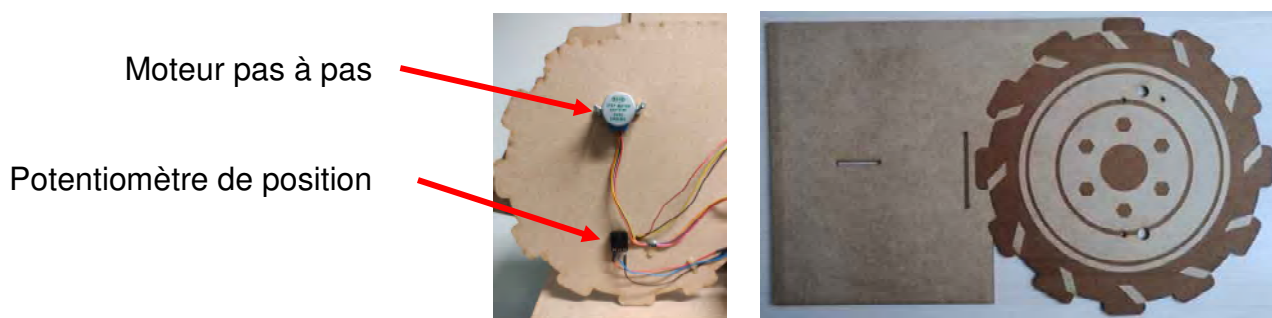


Vue arrière

Cette maquette a été entièrement réalisée avec du médium de 3 et 6mm et du plexiglass de 3mm, elle a été découpée à l'aide d'une découpe laser et est constituée de plusieurs parties à assembler.

Il faut préciser qu'il est possible de réaliser ce support pédagogique sans découpe laser, en réalisant seulement la partie opérative d'un système d'attelage trois points.

Il y a tout d'abord le fond où se trouve la gravure de la roue et sur laquelle sera fixé sur l'arrière le moteur pas à pas et le potentiomètre de position.

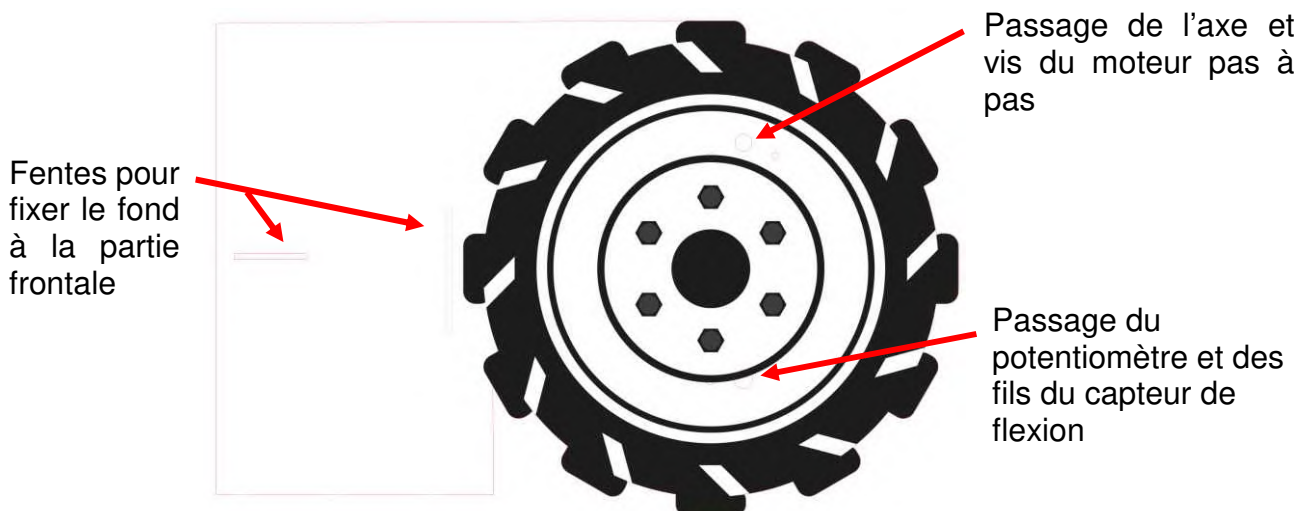


Moteur pas à pas

Potentiomètre de position

Moteur du relevage et capteur de position

Découpe de la roue



Passage de l'axe et vis du moteur pas à pas

Fentes pour fixer le fond à la partie frontale

Passage du potentiomètre et des fils du capteur de flexion

Schéma de la roue avec les différentes découpes

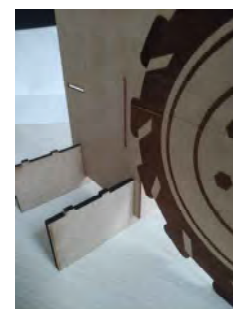
Ce panneau de la roue est emboîté dans les fentes de deux supports de 6 mm qui sont eux même fixés sur le plateau



Plan du découpage des supports

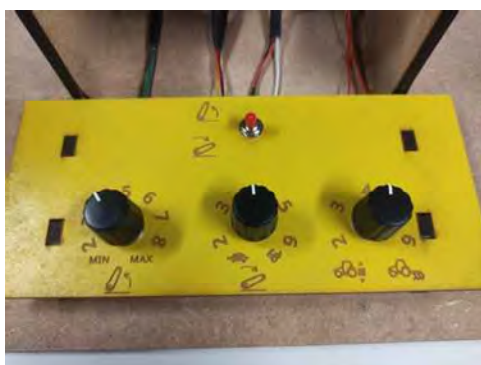


Supports découpés

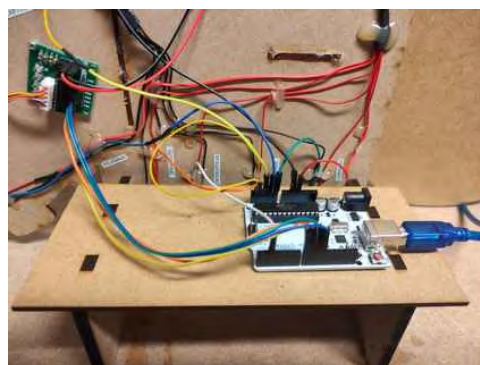


Supports en place

Sur ces deux supports de 6 mm sont aussi fixés à l'avant le pupitre et à l'arrière la plaque pour l'Arduino.



Vue du pupitre de commande



Vue de la plaque où se fixe l'Arduino

La face avant est fixée sur la roue à l'aide de deux morceaux de médium rectangulaire placés dans les fentes prévues à cet effet (et collées pour plus de solidité).



Découpe et gravure du tracteur



Vue des fixations des deux parties

Rectangle de médium de 3 mm collé dans les fentes, une verticale et l'autre horizontale (non visible sur la photo)

Enfin, le relevage est placé comme ceci : Le bras inférieur est rentré en force dans le potentiomètre de position. Les deux pivots sont fixés avec un tourillon de 8 mm et des goupilles.

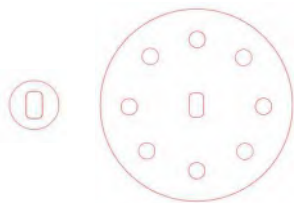


Les trois pièces du relevage



Relevage en place

Le bras du haut est fixé sur le moteur pas à pas à l'aide des pièces en plexiglass de 3mm ci-dessous :



plan du lien moteur/bras

La première est découpée trois fois et collée en les superposant, cela donnera un espace entre le moteur et le bras. Cet ensemble sera collé à la seconde pièce (en veillant à ce que l'axe soit respecté), et le tout sera vissé sur le bras à l'aide de vis placées dans les trous prévus.



Fixation sur bras supérieur



Vue de la jauge de contrainte

En dernier point, un élément de médium fendu (ou plusieurs collés ensemble) sera fixé sur le bras inférieur afin de pouvoir y glisser la jauge de contrainte.

## 4. Réalisation de la partie commande et de la partie opérative

### Etape 1 : Montée et descente à la consigne

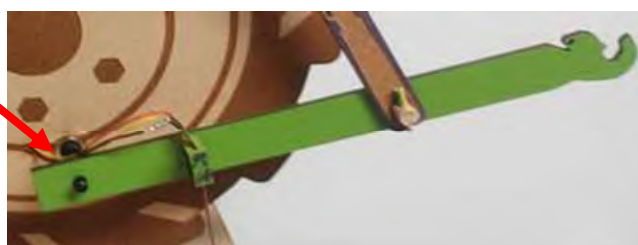
Dans un premier temps nous allons commencer par démarrer la maquette simplement, nous allons utiliser :

- un moteur pas à pas (28BYJ-48) classique pour lever ou descendre le bras du relevage. Il sera positionné sur l'axe du bras de relevage (en rouge sur la photo) et fixé à l'aide d'une pièce découpée dans du plexiglass pour la solidité et la fixation au bras.

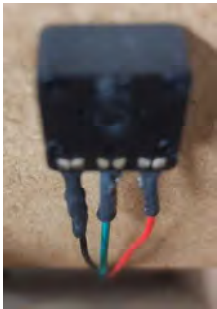
*Attention le moteur pas à pas 28BYJ-48 n'a pas un couple important, il ne faut donc pas que le système de relevage soit trop lourd pour qu'il puisse fonctionner et ne pas descendre sous son propre poids.*



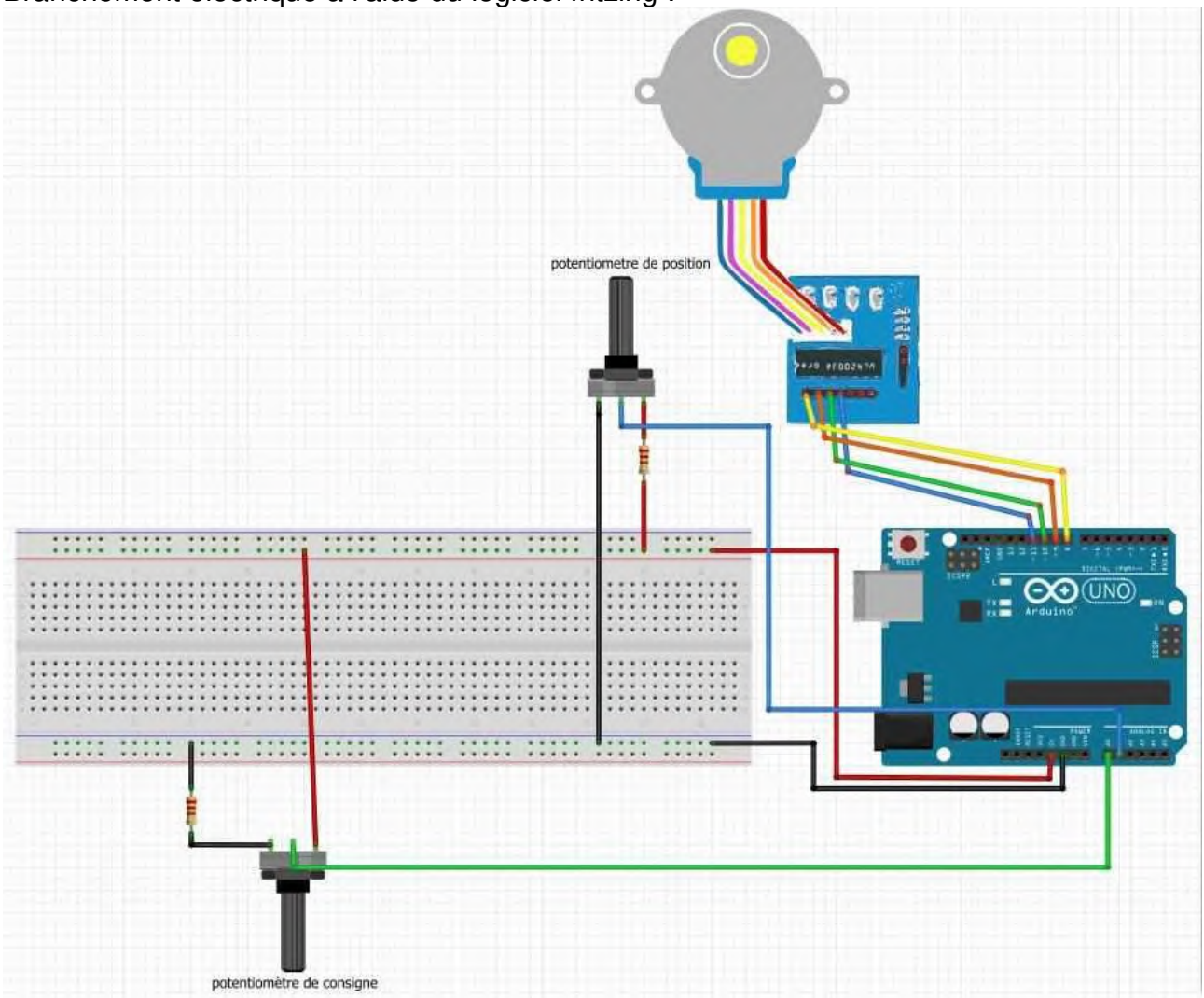
- un potentiomètre 10K Ohms placé sur l'axe du bras inférieur afin de connaître à tout instant la position des bras inférieurs.



- un premier potentiomètre de 10K Ohms sur le pupitre de commande pour donner la consigne.



Branchement électrique à l'aide du logiciel fritzing :



## Programme Arduino de l'étape 1 :

```
#include <Stepper.h> // charge la librairie qui pilote le moteur pas à pas
//cablage moteur pas à pas 28BYJ-48
//Pin8 au IN1 du driver ULN2003
//Pin9 au IN2 du driver ULN2003
//Pin10 au IN3 du driver ULN2003
//Pin11 au IN4 du driver ULN2003
const int Pas_par_tours = 32*64; //nombre de pas du moteur(on peut mettre 64*64 sur ce moteur pour plus
de précision)
Stepper monmoteur = Stepper(Pas_par_tours, 9, 11, 10, 8); //création de l'objet 'monmoteur' (nombre de
pas,1N2,1N4,1N3,1N1)

int potentiometre_consigne=A0; //potentiomètre de consigne branché sur l'entrée analogique A0
int potentiometre_relevage=A1; //potentiomètre de position du relevage branché sur A1
int consigne; //variable recevant la valeur de la consigne donnée manuellement par l'utilisateur
int relevage; //variable recevant la valeur de la position du relevage
int difference; //variable qui recevra la différence entre la consigne de profondeur et la position

void setup() {
  Serial.begin(9600); //ouverture du port série
  pinMode(potentiometre_consigne,INPUT);//déclaration de A0 en entrée
  pinMode(potentiometre_relevage,INPUT);//déclaration de A1 en entrée
}

void loop() {
  consigne=analogRead(potentiometre_consigne); //lit la valeur et la stocke dans la variable "consigne"
(valeur entre 0 et 1023)
  consigne = map(consigne,0,1023,700,900); // met l'amplitude de la consigne à l'échelle de la course
souhaitée du relevage (700 valeur du relevage bas et 900 haut)
  relevage=analogRead(potentiometre_relevage); //lit la valeur et la stocke dans la variable "relevage"
(valeur entre 0 et 1023)
  Serial.print(consigne); //écrit sur le port série la valeur de la variable consigne
  Serial.print(" "); //écrit sur le port série des espaces
  Serial.println(relevage); //écrit sur le port série la valeur de la variable relevage et revient à la ligne
  difference = consigne-relevage;//stocke dans la variable "difference" la soustraction
  if (difference>5){ //si cette différence est supérieure à 5
    monmoteur.setSpeed(5); //vitesse du moteur
    monmoteur.step(-1); //on fait tourner le moteur de 1 pas pour baisser (sens horaire)
  }
  if(difference<-5){ //si cette différence est inférieure à -5
    monmoteur.setSpeed(5); //vitesse du moteur
    monmoteur.step(1); //on fait tourner le moteur de 1 pas pour lever (sens anti-horaire)
  }
  delay(50);//attends 50 millisecondes, cette ligne peut-être supprimée pour une meilleure rapidité de
réaction du relevage
}
```

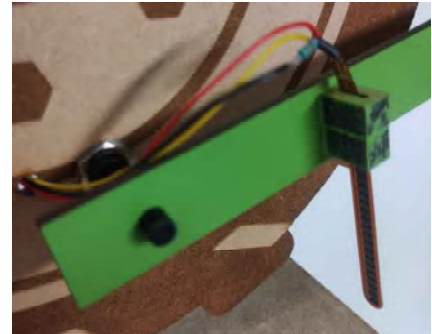
Attention les valeurs 700 et 900 pour la consigne du relevage dépend de votre réglage du potentiomètre fixé sur le bras de relevage, elles seront à déterminer en fonction de votre maquette et de votre branchement électrique (le sens peut être inversé).

A ce stade vous avez déjà un système de relevage opérationnel, qui monte ou descend en fonction de la consigne que vous lui donnez, donc un système de contrôle de la position.



## **Etape 2 : ajustement de la profondeur en fonction de la résistance sur le capteur de flexion**

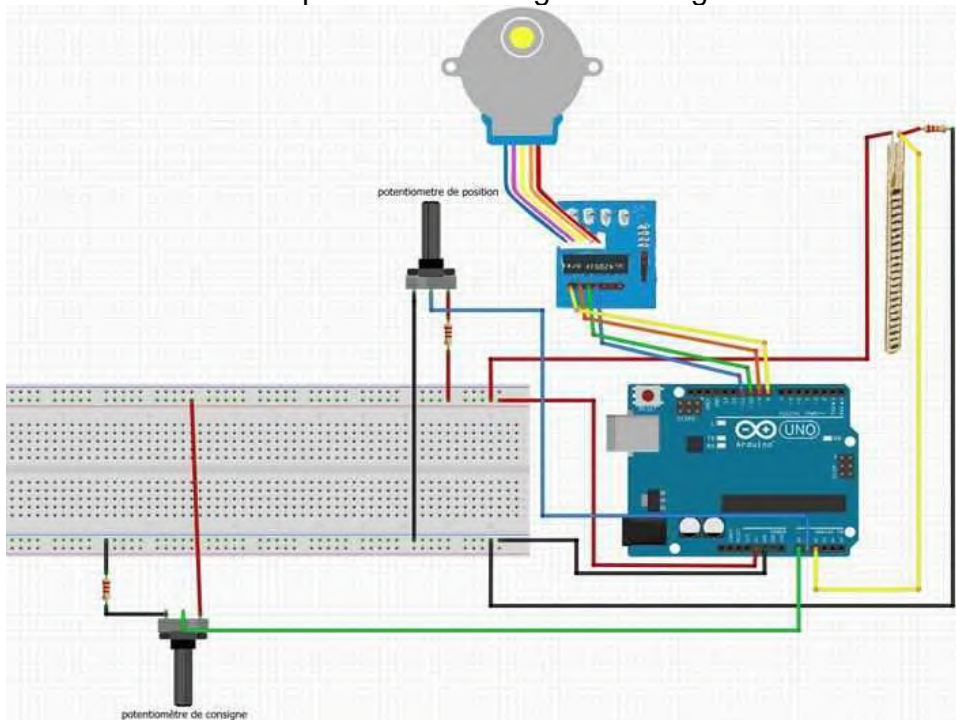
Nous allons ajouter au montage un capteur de flexion (fs7954-20718). Celui-ci permettra de simuler, lorsqu'on le touchera avec la main, une résistance du sol sur l'outil. Si une résistance est appliquée, le relevage devra monter, et si la résistance diminue, le relevage devra reprendre sa position de consigne. Tout cela afin que le tracteur garde une vitesse constante dans la réalité.



Vue de la jauge de contrainte

Ce capteur de flexion est ajouté au bout du bras inférieur

Branchement électrique à l'aide du logiciel fritzing :



Programme de l'étape 2 (les nouveaux éléments sont **surligné en jaune**) :

```
#include <Stepper.h> // charge la librairie qui pilote le moteur pas à pas
//cablage moteur pas à pas 28BYJ-48
//Pin8 au IN1 du driver ULN2003
//Pin9 au IN2 du driver ULN2003
//Pin10 au IN3 du driver ULN2003
//Pin11 au IN4 du driver ULN2003
const int Pas_par_tours = 32*64; //nombre de pas du moteur(on peut mettre 64*64 sur ce moteur pour plus
de précision)
Stepper monmoteur = Stepper(Pas_par_tours, 9, 11, 10, 8); //création de l'objet 'monmoteur' (nombre de
pas,1N2,1N4,1N3,1N1)
```

```

int potentiometre_consigne=A0; //potentiomètre de consigne branché sur l'entrée analogique A0
int potentiometre_relevage=A1; //potentiomètre de position du relevage branché sur A1
int capteur_flexion=A2; //capteur de flexion branché sur A2
long valeur_flexion; //variable recevant la valeur du capteur de flexion
int consigne; //variable recevant la valeur de la consigne donnée manuellement par l'utilisateur
int relevage; //variable recevant la valeur de la position du relevage
int profondeur; // variable qui recevra la consigne mis à l'échelle de l'influence de la flexion
int difference; //variable qui recevra la différence entre la consigne de profondeur et la position

void setup() {
  Serial.begin(9600); //ouverture du port série
  pinMode(potentiometre_consigne,INPUT);//déclaration de A0 en entrée
  pinMode(potentiometre_relevage,INPUT);//déclaration de A1 en entrée
  pinMode(capteur_flexion,INPUT);//déclaration de A2 en entrée
}

void loop() {
  consigne=analogRead(potentiometre_consigne); //lit la valeur et la stocke dans la variable "consigne"
  (valeur entre 0 et 1023)
  consigne = map(consigne,0,1023,700,900); // met l'amplitude de la consigne à l'échelle de la course
  souhaitée du relevage (700 valeur du relevage bas et 900 haut)
  relevage=analogRead(potentiometre_relevage); //lit la valeur et la stocke dans la variable "relevage"
  (valeur entre 0 et 1023)
  Serial.print(consigne); //écrit sur le port série la valeur de la variable consigne
  Serial.print(" "); //écrit sur le port série des espaces
  Serial.print(relevage); //écrit sur le port série la valeur de la variable relevage
  Serial.print(" ");
  valeur_flexion=analogRead(capteur_flexion); //lit la valeur et la stocke dans la variable "valeur_flexion"
  Serial.print(valeur_flexion); //écrit sur le port série la valeur de la variable

  /* Nous allons maintenant stocker dans "profondeur" la mise à l'échelle de la variable "valeur_flexion" ce
  qui donne
  - la valeur de la consigne lorsque le capteur est au repos(valeur 450)
  - la valeur 700 (valeur du relevage en position haute) lorsque le capteur est sous flexion maximale(330)
  */
  profondeur=map(valeur_flexion,450,330,consigne,700); //transforme" 450 en la valeur de la consigne et
  330 en 700 (max du relevage)
  Serial.print("  profondeur"); //écrit sur le port série
  Serial.println(profondeur); //écrit sur le port série la valeur de la variable et revient à la ligne

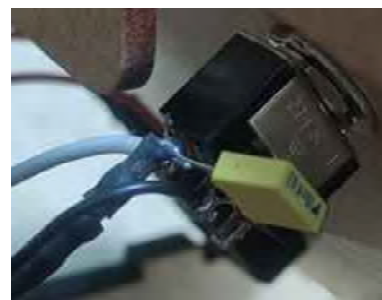
  difference= profondeur-relevage;//stocke dans "difference" la soustraction des valeurs
  if (difference>5){ //si cette différence est supérieure à 5
    monmoteur.setSpeed(5); //vitesse du moteur
    monmoteur.step(-1); //on fait tourner le moteur de 1 pas pour baisser (sens horaire)
  }
  if(difference<-5){ //si cette différence est inférieure à -5
    monmoteur.setSpeed(5); //vitesse du moteur
    monmoteur.step(1); //on fait tourner le moteur de 1 pas pour lever (sens anti-horaire)
  }
  delay(50);//attends 50 millisecondes, cette ligne peut être supprimée pour une meilleure rapidité de
  réaction du relevage
}

```

### Etape 3 : ajout d'un bouton de commande pour actionner le relevage en position haute ou basse

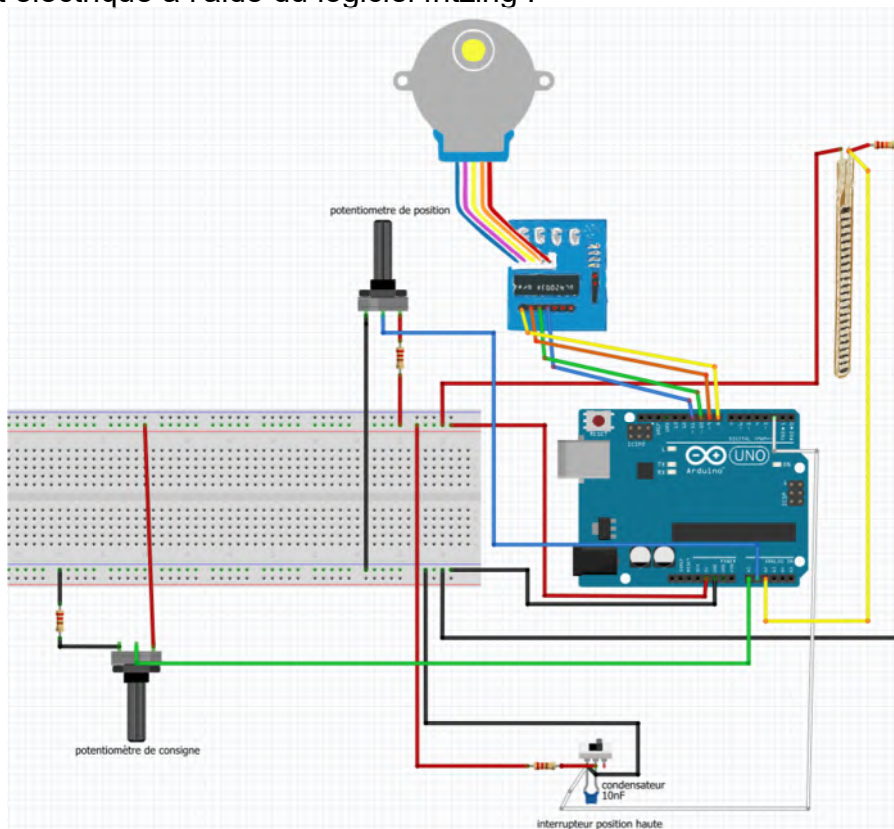
Nous allons ajouter un interrupteur qui, lorsqu'il sera actionné, mettra directement le relevage en position haute, par exemple pour tourner en bout de rangs.

Un condensateur de 10 nF est ajouté aux bornes de l'interrupteur pour éviter les effets de rebonds.



Vue de l'interrupteur avec condensateur

Branchement électrique à l'aide du logiciel fritzing :



Programme de l'étape 3 :

```
#include <Stepper.h> // charge la librairie qui pilote le moteur pas à pas
//cablage moteur pas à pas 28BYJ-48
//Pin8 au IN1 du driver ULN2003
//Pin9 au IN2 du driver ULN2003
//Pin10 au IN3 du driver ULN2003
//Pin11 au IN4 du driver ULN2003
const int Pas_par_tours = 32*64; //nombre de pas du moteur(on peut mettre 64*64 sur ce moteur pour plus
de précision)
Stepper monmoteur = Stepper(Pas_par_tours, 9, 11, 10, 8); //création de l'objet 'monmoteur' (nombre de
pas,1N2,1N4,1N3,1N1)

int potentiometre_consigne=A0; //potentiomètre de consigne branché sur l'entrée analogique A0
int potentiometre_relevage=A1; //potentiomètre de position du relevage branché sur A1
int capteur_flexion=A2; //capteur de flexion branché sur A2
int bouton=2; //interrupteur de choix d'activation ou non(du relevage position haute)
bool etatbouton; //etat logique du bouton (0 ou 1)
long valeur_flexion; //variable recevant la valeur du capteur de flexion
int consigne; //variable recevant la valeur de la consigne donnée manuellement par l'utilisateur
```

```

int releavage; //variable recevant la valeur de la position du releavage
int profondeur; // variable qui recevra la consigne mis à l'échelle de l'influence de la flexion
int difference; //variable qui recevra la difference entre la consigne de profondeur et la position

void setup() {
  Serial.begin(9600); //ouverture du port série
  pinMode(potentiometre_consigne,INPUT);//déclaration de A0 en entrée
  pinMode(potentiometre_relevage,INPUT);//déclaration de A1 en entrée
  pinMode(capteur_flexion,INPUT);//déclaration de A2 en entrée
  pinMode(bouton,INPUT); //déclaration de la pin2 en entrée
}

void loop() {
  etatbouton=digitalRead(bouton);

  consigne=analogRead(potentiometre_consigne); //lit la valeur et la stocke dans la variable "consigne"
  (valeur entre 0 et 1023)
  consigne = map(consigne,0,1023,700,900); // met l'amplitude de la consigne à l'échelle de la course
  souhaitée du releavage (700 valeur du releavage bas et 900 haut)
  releavage=analogRead(potentiometre_relevage); //lit la valeur et la stocke dans la variable "relevage"
  (valeur entre 0 et 1023)
  Serial.print(consigne); //écrit sur le port série la valeur de la variable consigne
  Serial.print(" "); //écrit sur le port série des espaces
  Serial.print(relevage); //écrit sur le port série la valeur de la variable releavage
  Serial.print(" ");

  if (etatbouton==1) {
    while(relevage>700){
      releavage=analogRead(potentiometre_relevage); //on relit la valeur pour sortir de la boucle au bon
      moment
      monmoteur.setSpeed(4); //vitesse du moteur
      monmoteur.step(1); //on fait tourner le moteur de 1 pas pour lever (sens anti-horaire)
    }
  }

  else{
    valeur_flexion=analogRead(capteur_flexion); //lit la valeur et la stocke dans la variable "valeur_flexion"
    Serial.print(valeur_flexion); //écrit sur le port série la valeur de la variable

    /* Nous allons maintenant stocker dans "profondeur" la mise à l'échelle de la variable "valeur_flexion" ce
    qui donne
    - la valeur de la consigne lorsque le capteur est au repos(valeur 450)
    - la valeur 700 (valeur du releavage en position haute) lorsque le capteur est sous flexion maximale(330)
    */
    profondeur=map(valeur_flexion,450,330,consigne,700); //"transforme" 450 en la valeur de la consigne et
    330 en 700 (max du releavage)
    Serial.print("  profondeur"); //écrit sur le port série
    Serial.println(profondeur); //écrit sur le port série la valeur de la variable et revient à la ligne

    difference= profondeur-relevage;//stocke dans "difference" la soustraction des valeurs
    if (difference>5){ //si cette différence est supérieure à 5
      monmoteur.setSpeed(5); //vitesse du moteur
      monmoteur.step(-1); //on fait tourner le moteur de 1 pas pour baisser(sens horaire)
    }
    if(difference<-5){ //si cette différence est inférieure à -5
      monmoteur.setSpeed(5); //vitesse du moteur
      monmoteur.step(1); //on fait tourner le moteur de 1 pas pour lever(sens anti-horaire)
    }
  }
}

```

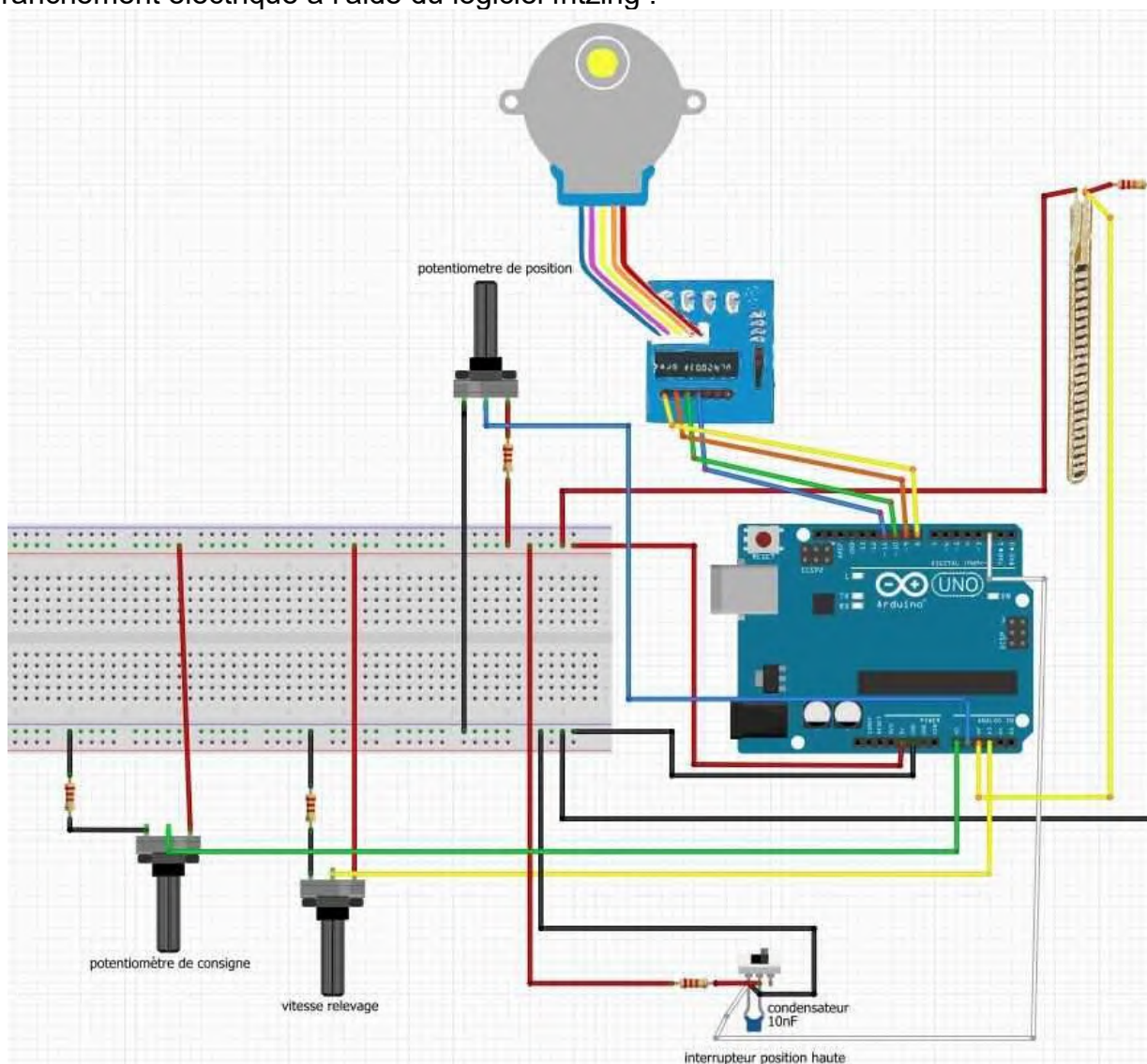
#### Etape 4 : réglage de la vitesse (montée et descente) du système de relevage en fonction d'un potentiomètre de consigne

Nous allons maintenant ajouter le réglage de la vitesse de montée ou descente du système de relevage à l'aide d'un potentiomètre de consigne. Nous allons donc ajouter un potentiomètre de 10KOhms sur le pupitre de commande.



Vue du pupitre avec la commande de vitesse

Branchement électrique à l'aide du logiciel fritzing :



## Programme de l'étape 4 :

```
#include <Stepper.h> // charge la librairie qui pilote le moteur pas à pas
//cablage moteur pas à pas 28BYJ-48
//Pin8 au IN1 du driver ULN2003
//Pin9 au IN2 du driver ULN2003
//Pin10 au IN3 du driver ULN2003
//Pin11 au IN4 du driver ULN2003
const int Pas_par_tours = 32*64; //nombre de pas du moteur(on peut mettre 64*64 sur ce moteur pour plus
de précision)
Stepper monmoteur = Stepper(Pas_par_tours, 9, 11, 10, 8); //création de l'objet 'monmoteur' (nombre de
pas,1N2,1N4,1N3,1N1)

int potentiometre_consigne=A0; //potentiomètre de consigne branché sur l'entrée analogique A0
int potentiometre_relevage=A1; //potentiomètre de position du relevage branché sur A1
int capteur_flexion=A2; //capteur de flexion branché sur A2
int capteur_vitesse=A3; // potentiomètre de consigne de la vitesse
int bouton=2; //interrupteur de choix d'activation ou non(du relevage position haute)
bool etatbouton; //etat logique du bouton (0 ou 1)
long valeur_flexion; //variable recevant la valeur du capteur de flexion
int valeur_vitesse; //variable qui recevra la valeur de la vitesse
int vitesse; // variable qui recevra la vitesse mise à m'échelle(1 à 6)
int consigne; //variable recevant la valeur de la consigne donnée manuellement par l'utilisateur
int relevage; //variable recevant la valeur de la position du relevage
int profondeur; // variable qui recevra la consigne mis à l'échelle de l'influence de la flexion
int difference; //variable qui recevra la difference entre la consigne de profondeur et la position

void setup() {
  Serial.begin(9600); //ouverture du port série
  pinMode(potentiometre_consigne,INPUT);//déclaration de A0 en entrée
  pinMode(potentiometre_relevage,INPUT);//déclaration de A1 en entrée
  pinMode(capteur_flexion,INPUT);//déclaration de A2 en entrée
  pinMode(bouton,INPUT); //déclaration de 2 en entrée
  pinMode(capteur_vitesse, INPUT); //déclaration de A3 en entrée
}

void loop() {

  etatbouton=digitalRead(bouton); //met l'état du bouton (0 ou 1) dans la variable

  valeur_vitesse=analogRead(capteur_vitesse); //lit la valeur et la stocke dans la variable "valeur vitesse"
(valeur entre 0 et 1023)
  vitesse= map(valeur_vitesse,0,1023,1,6); //change l'échelle de vitesse de 1024 valeurs en 6 valeurs
  Serial.print(vitesse); //écrit la vitesse
  Serial.print(" "); // écrit 3 espaces
  consigne=analogRead(potentiometre_consigne); //lit la valeur et la stocke dans la variable "consigne"
(valeur entre 0 et 1023)
  consigne = map(consigne,0,1023,700,900); // met l'amplitude de la consigne à l'échelle de la course
souhaitée du relevage (700 valeur du relevage bas et 900 haut)
  relevage=analogRead(potentiometre_relevage); //lit la valeur et la stocke dans la variable "relevage"
(valeur entre 0 et 1023)
  Serial.print(consigne); //écrit sur le port série la valeur de la variable consigne
  Serial.print(" "); //écrit sur le port série des espaces
  Serial.print(relevage); //écrit sur le port série la valeur de la variable relevage
  Serial.print(" ");

  if (etatbouton==1) {
    while(relevage>700){
      relevage=analogRead(potentiometre_relevage); //on relit la valeur pour sortir de la boucle au bon
moment
      valeur_vitesse=analogRead(capteur_vitesse); //lit la valeur et la stocke dans la variable "valeur vitesse"
(valeur entre 0 et 1023)
      vitesse= map(valeur_vitesse,0,1023,1,6); //change l'échelle de vitesse de 1024 valeurs en 6 valeurs
```

```

    monmoteur.setSpeed(vitesse); //vitesse du moteur
    monmoteur.step(1); //on fait tourner le moteur de 1 pas pour lever (sens anti-horaire)
}
}

else{
    valeur_flexion=analogRead(capteur_flexion); //lit la valeur et la stocke dans la variable "valeur_flexion"
    Serial.print(valeur_flexion); //écrit sur le port série la valeur de la variable

    /* Nous allons maintenant stocker dans "profondeur" la mise à l'échelle de la variable "valeur_flexion" ce
    qui donne
    - la valeur de la consigne lorsque le capteur est au repos(valeur 450)
    - la valeur 700 (valeur du relevage en position haute) lorsque le capteur est sous flexion maximale (330)
    */
    profondeur=map(valeur_flexion,450,330,consigne,700); //"transforme" 450 en la valeur de la consigne et
    330 en 700 (max du relevage)
    Serial.print("  profondeur"); //écrit sur le port série
    Serial.println(profondeur); //écrit sur le port série la valeur de la variable et revient à la ligne

    difference= profondeur-relevage;//stocke dans "difference" la soustraction des valeurs
    if (difference>5){ //si cette différence est supérieure à 5
        monmoteur.setSpeed(vitesse); //vitesse du moteur
        monmoteur.step(-5); //on fait tourner le moteur de 5 pas pour baisser(sens horaire)
    }
    if(difference<-5){ //si cette différence est inférieure à -5
        monmoteur.setSpeed(vitesse); //vitesse du moteur
        monmoteur.step(5); //on fait tourner le moteur de 5 pas pour lever(sens anti-horaire)
    }
}
}
}
}

```

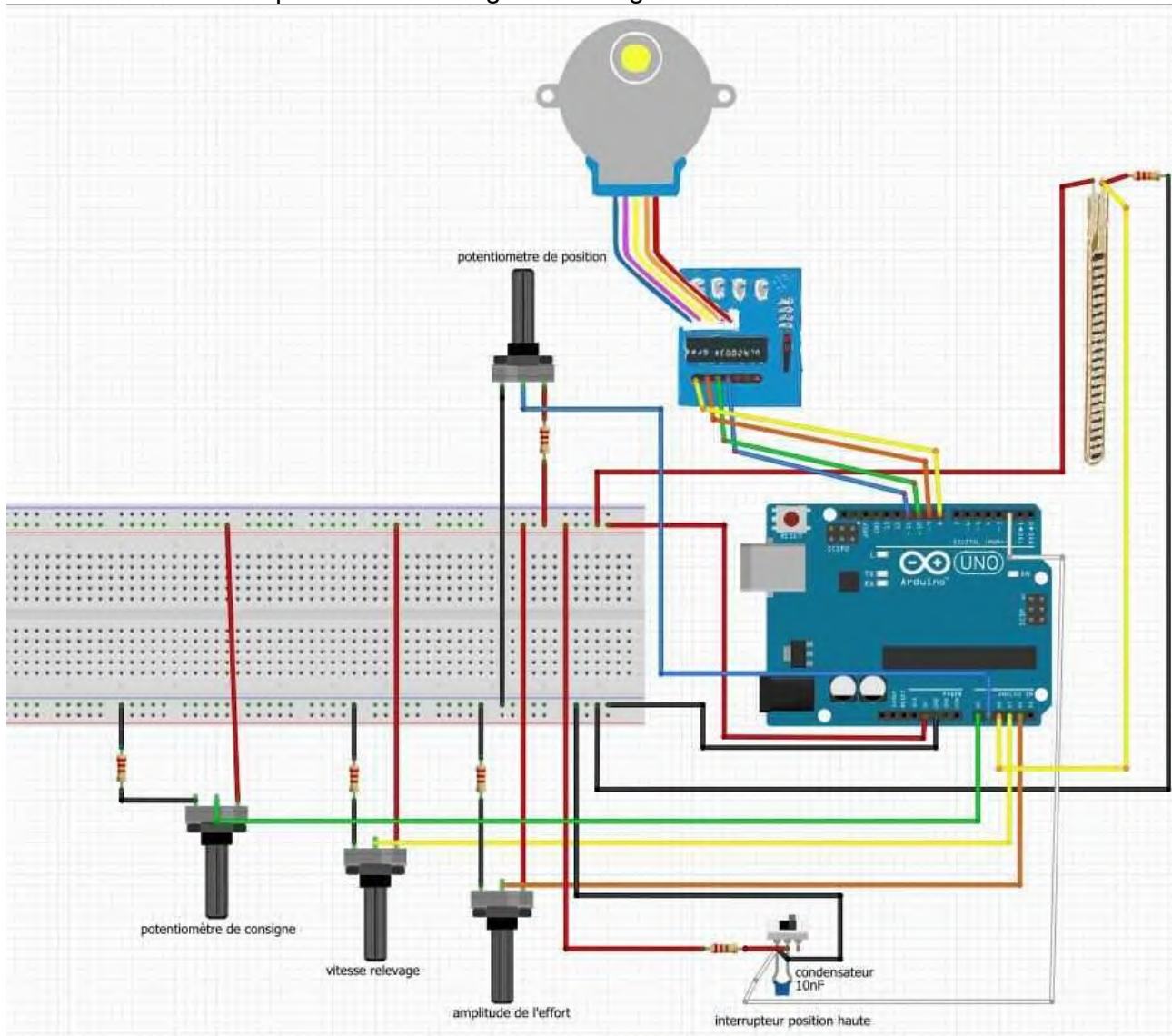
### **Etape 5 : Système de contrôle d'effort de traction**

Nous allons ajouter un dernier potentiomètre de 10 KOhms sur le pupitre de commande. Celui-ci aura pour but de régler la sensibilité du contrôle d'effort, c'est-à-dire que si nous réglons le contrôle très sensible, le relevage montera plus haut que si nous le réglons peu sensible. L'effort peut ainsi varier plus rapidement.



*Vue du pupitre avec la commande sensibilité de l'effort*

## Branchement électrique à l'aide du logiciel fritzing :



## Programme étape 5 final :

```
#include <Stepper.h> // charge la librairie qui pilote le moteur pas à pas
//cablage moteur pas à pas 28BYJ-48
//Pin8 au IN1 du driver ULN2003
//Pin9 au IN2 du driver ULN2003
//Pin10 au IN3 du driver ULN2003
//Pin11 au IN4 du driver ULN2003
const int Pas_par_tours = 32*64; //nombre de pas du moteur(on peut mettre 64*64 sur ce moteur pour plus
de précision)
Stepper monmoteur = Stepper(Pas_par_tours, 9, 11, 10, 8); //création de l'objet 'monmoteur' (nombre de
pas,1N2,1N4,1N3,1N1)

int potentiometre_consigne=A0; //potentiometre de consigne branché sur l'entrée analogique A0
int potentiometre_relevage=A1; //potentiometre de position du relevage branché sur A1
int capteur_flexion=A2; //capteur de flexion branché sur A2
int potentiometre_vitesse=A3; // potentiometre de consigne de la vitesse
int potentiometre_sensibilite_effort=A4; // potentiometre de consigne de sensibilité de la contrainte d'effort
int bouton=2; //interrupteur de choix d'activation ou non(du relevage position haute)
bool etatbouton; //etat logique du bouton (0 ou 1)
long valeur_flexion; //variable recevant la valeur du capteur de flexion(456 au repos, 330 avec forte flexion)
int valeur_vitesse; //variable qui recevra la valeur de la vitesse
int vitesse; // variable qui recevra la vitesse mise à m'échelle(0 à 6)
```



```

int consigne; //variable recevant la valeur de la consigne donnée manuellement par l'utilisateur
int relevage; //variable recevant la valeur de la position du relevage
int profondeur; // variable qui recevra la consigne mis à l'échelle de l'influence de la flexion
int difference; //variable qui recevra la différence entre la consigne de profondeur et la position
int valeur_sensibilite_effort; // variable qui recevra la valeur du potentiomètre sensibilité-effort
int sensibilite_effort; // variable qui recevra la sensibilité mis à l'échelle (0 à 5)
int flex_max; // variable qui recevra la valeur max de la flexion en fonction de la sensibilité réglée

void setup() {
  Serial.begin(9600); //ouverture du port série
  pinMode(potentiometre_consigne,INPUT);//déclaration de A0 en entrée
  pinMode(potentiometre_relevage,INPUT);//déclaration de A1 en entrée
  pinMode(capteur_flexion,INPUT);//déclaration de A2 en entrée
  pinMode(bouton,INPUT); //déclaration de 2 en entrée
  pinMode(potentiometre_vitesse,INPUT); //déclaration de A3 en entrée
  pinMode(potentiometre_sensibilite_effort,INPUT); //déclaration de A4 en entrée
}

void loop() {
  etatbouton=digitalRead(bouton); //met l'état du bouton (0 ou 1) dans la variable

  valeur_vitesse=analogRead(potentiometre_vitesse); //lit la valeur et la stocke dans la variable "valeur
vitesse" (valeur entre 0 et 1023)
  vitesse= map(valeur_vitesse,0,1023,1,7); //change l'échelle de vitesse de 1024 valeurs en 6 valeurs (de 1
à 6)
  Serial.print(vitesse); //écrit la vitesse
  Serial.print(" "); // écrit 3 espaces
  consigne=analogRead(potentiometre_consigne); //lit la valeur et la stocke dans la variable "consigne"
(valeur entre 0 et 1023)
  consigne = map(consigne,0,1023,720,900); // met l'amplitude de la consigne à l'échelle de la course
souhaitée du relevage (720 valeur du relevage haut et 900 bas)
  relevage=analogRead(potentiometre_relevage); //lit la valeur et la stocke dans la variable "relevage"
(valeur entre 0 et 1023)
  Serial.print(consigne); //écrit sur le port série la valeur de la variable consigne
  Serial.print(" "); //écrit sur le port série des espaces
  Serial.print(relevage); //écrit sur le port série la valeur de la variable relevage
  Serial.print(" ");

  if (etatbouton==1) {
    while(relevage>720){
      relevage=analogRead(potentiometre_relevage); //on relit la valeur pour sortir de la boucle au bon
moment
      valeur_vitesse=analogRead(potentiometre_vitesse); //lit la valeur et la stocke dans la variable "valeur
vitesse" (valeur entre 0 et 1023)
      vitesse= map(valeur_vitesse,0,1023,1,7); //change l'échelle de vitesse de 1024 valeurs en 6 valeurs (de
1 à 6)
      monmoteur.setSpeed(vitesse); //vitesse du moteur
      monmoteur.step(1); //on fait tourner le moteur de 1 pas pour lever (sens anti-horaire)
    }
  }

  else{
    valeur_flexion=analogRead(capteur_flexion); //lit la valeur et la stocke dans la variable "valeur_flexion"
    Serial.print(valeur_flexion); //écrit sur le port série la valeur de la variable
    valeur_sensibilite_effort=analogRead(potentiometre_sensibilite_effort); //lit la valeur(0 à 1023)et la
stocke dans la variable
    sensibilite_effort=map(valeur_sensibilite_effort,0,1023,1,7); // met à l'echelle (1 à 6)

    /* Nous allons maintenant stocker dans "profondeur" la mise à l'échelle de la variable "valeur_flexion" ce
qui donne
    - la valeur de la consigne lorsque le capteur est au repos(valeur 435)
    - la valeur 720 (valeur du relevage en position haute) lorsque le capteur est sous flexion maximale(300)
et nous allons faire varier cette valeur maximale en fonction de la sensibilité

```

