

Manuel technique

Important : Ce manuel est à disposition de tout utilisateur souhaitant effectuer des modifications sur le système d'alimentation de l'extrudeuse Noztek. Il regroupe l'ensemble des informations et des tests déjà réalisés sur le dispositif. Nous vous invitons à modifier ce document si des modifications sont apportées au système (ceci pour les prochains utilisateurs de la machine).

Problématique : Comment alimenter de façon autonome une extrudeuse en granulés plastiques recyclés pour la fabrication de filament d'impression 3D ?

Schémas du système

VUE INTERNE :

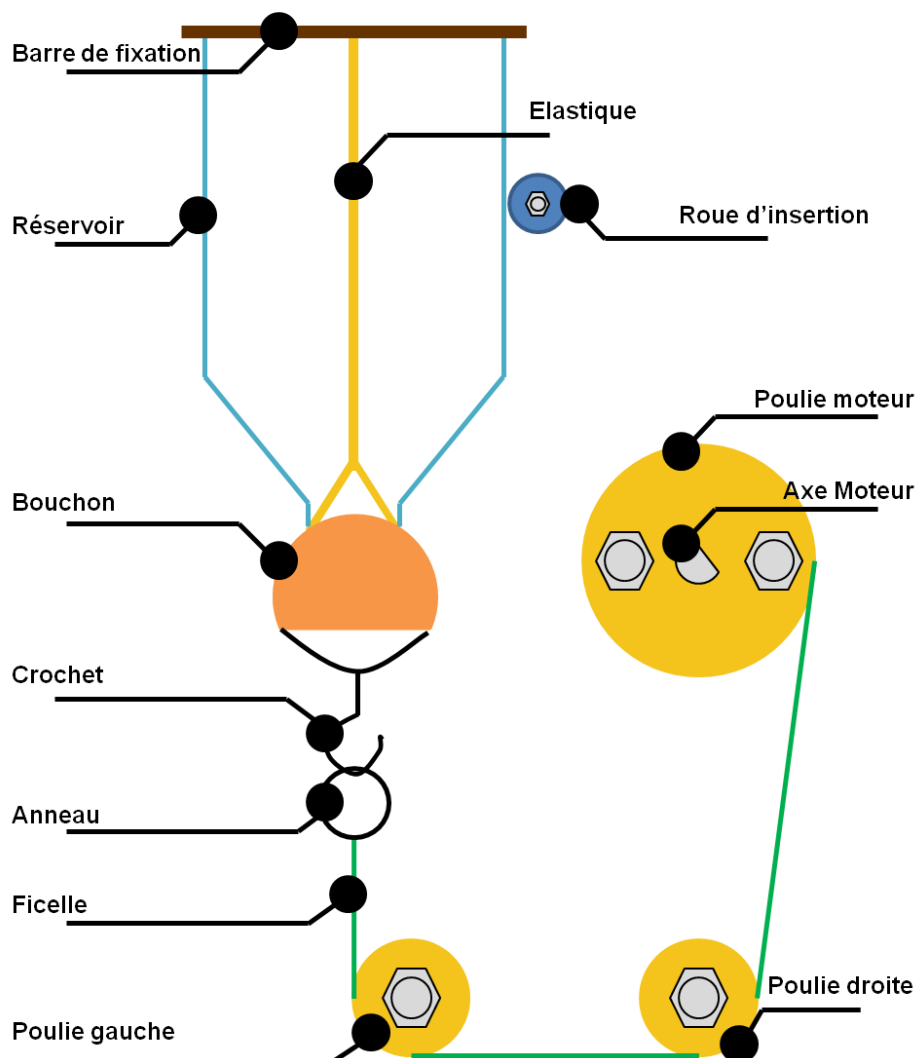


Figure 1 : Schéma interne système

VUE BOITIER :

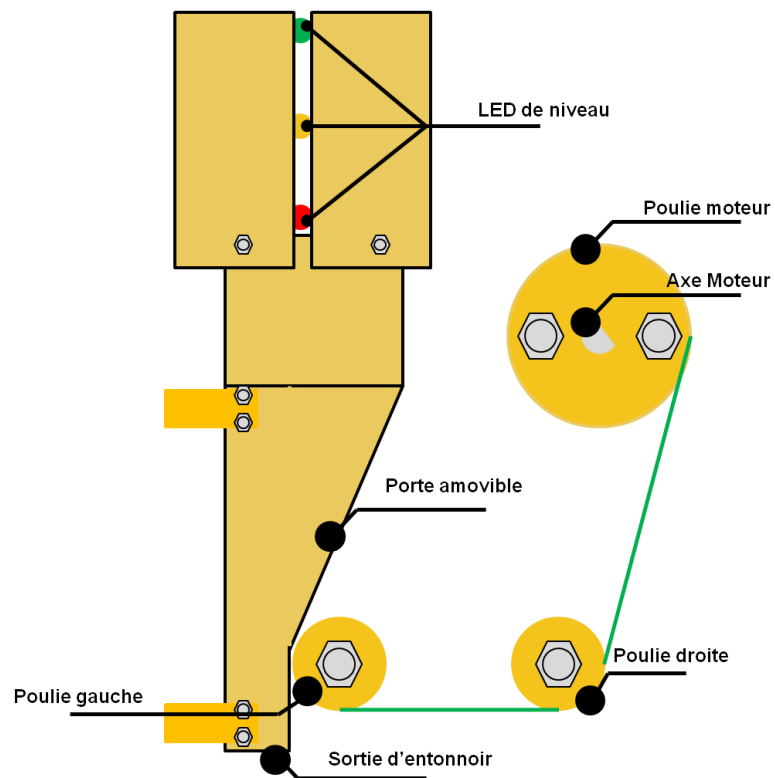


Figure 2 : Schéma externe système

BOITIER ELECTRONIQUE :

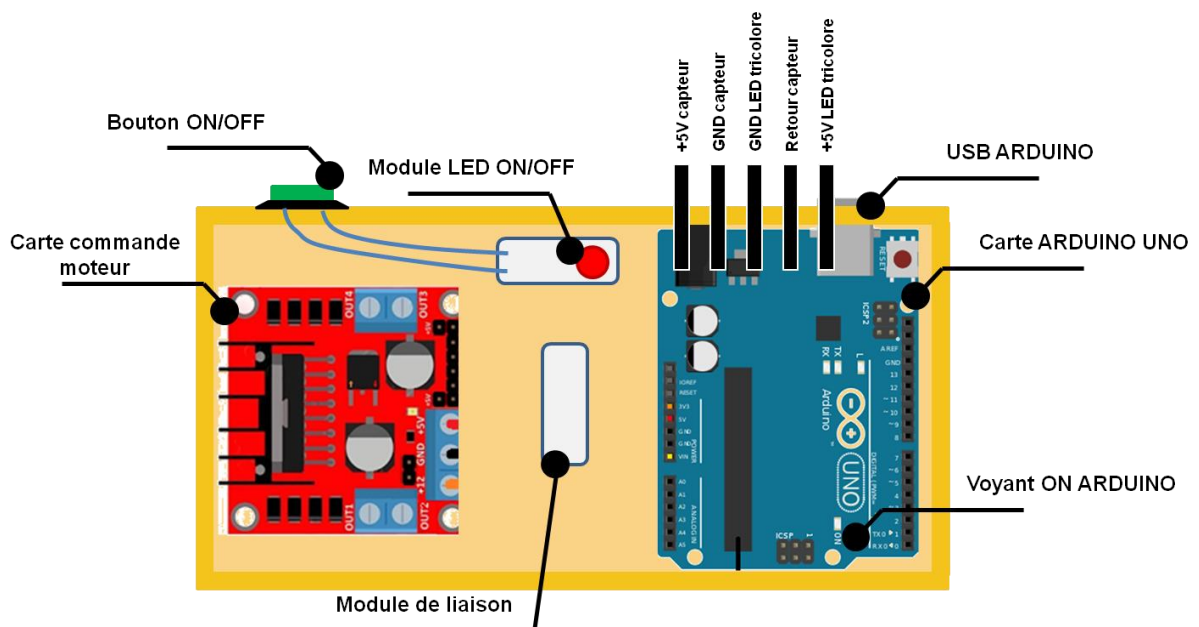


Figure 3 : Schéma carte électronique

ENTONNOIR EXTRUDEUSE

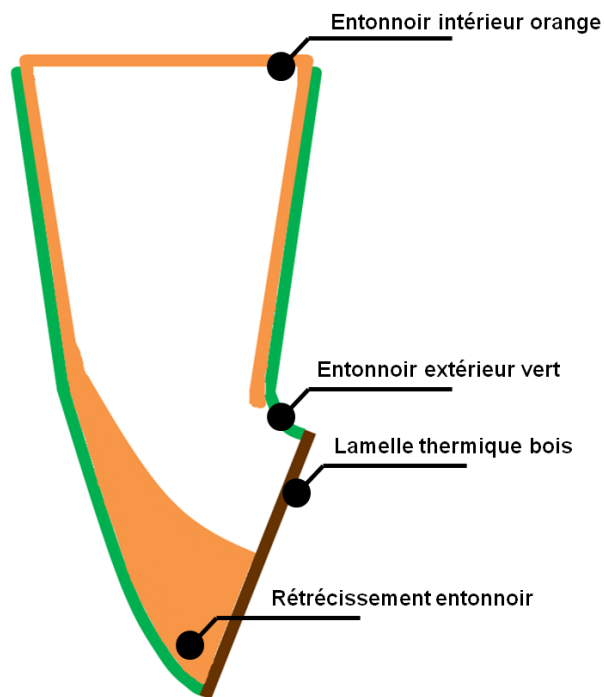


Figure 4 : Schéma entonnoir

Les schémas proposés ci-dessus ne sont pas conformes au système réel en termes de dimensions

Boitier **Verte** : Entonnoir principal pour insertion granules dans l'extrudeuse.

Boitier **Orange** : Entonnoir secondaire réduisant l'ouverture d'insertion pour éviter les bourrages au niveau de la vis d'extrusion.

Ces deux boitiers ont été créés mais n'ont pas pu être testés du fait que l'extrudeuse fut en panne durant la conception.

Principe de fonctionnement du système :

Le système final consiste en une trémie à ouverture actionnée par rotation d'un moteur (lui-même piloté par un boîtier électronique).

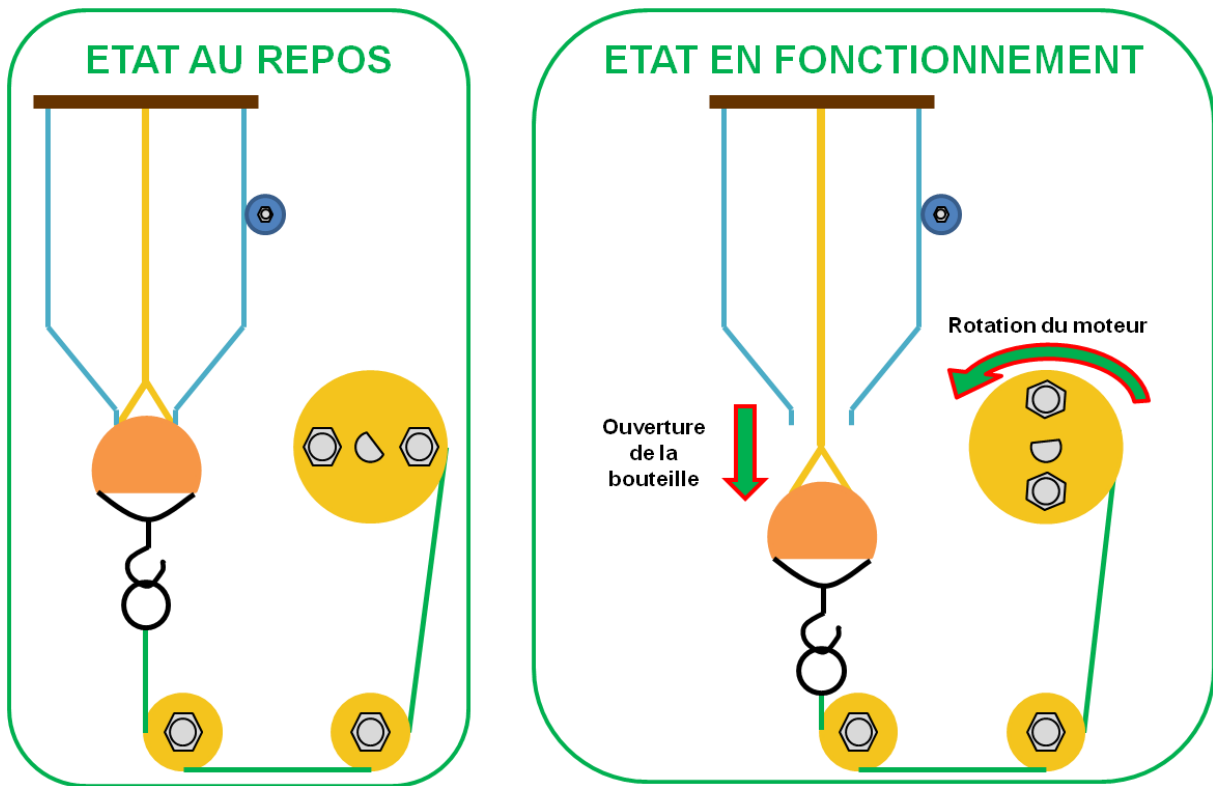


Figure 5 : Schéma mise en situation

Bouchon et poulies

Le système d'ouverture est constitué d'un bouchon retenu par un élastique. Lors de la rotation du moteur, l'élastique s'étire pour déverser le granule dans l'extrudeuse. L'idée des poulies consiste à déporter le moteur électrique pour ne pas le bloquer à cause de grain lors de l'ouverture.

Liste des éléments du système électronique

ARDUINO UNO

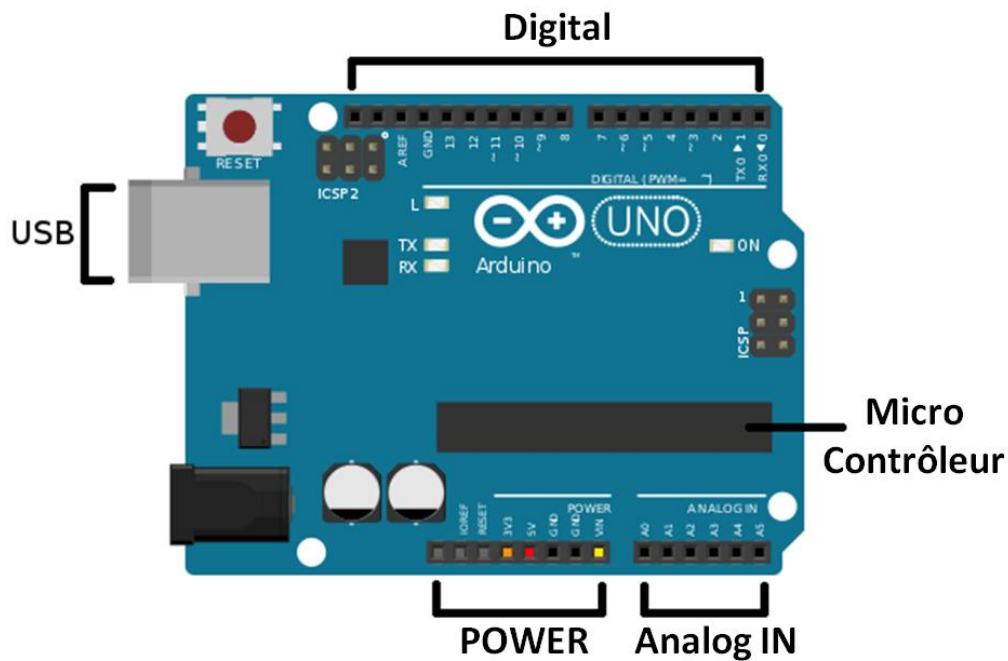


Figure 6 : Schéma ARDUINO UNO

Digital : Ces broches sont des sorties dont l'état peut varier entre « haut » et « bas ». Elles servent notamment à commander la rotation du moteur. Les broches noté PWM peuvent également servir d'alimentation si leurs états sont mis à « haut ».

USB : Cette prise femelle permet de connecter la carte ARDUINO au pc pour transférer le programme. Elle sert également à alimenter la carte ARDUINO.

MICRO Contrôleur : Ce composant électronique est le cerveau de la carte. Il contient le programme ARDUINO.

POWER : Différente broche permettant d'alimenter des modules annexes en 3,3V – 5V ou 12V

Analog IN : Entrée analogique permettant de récupérer l'état notamment de capteurs.

Module L298N

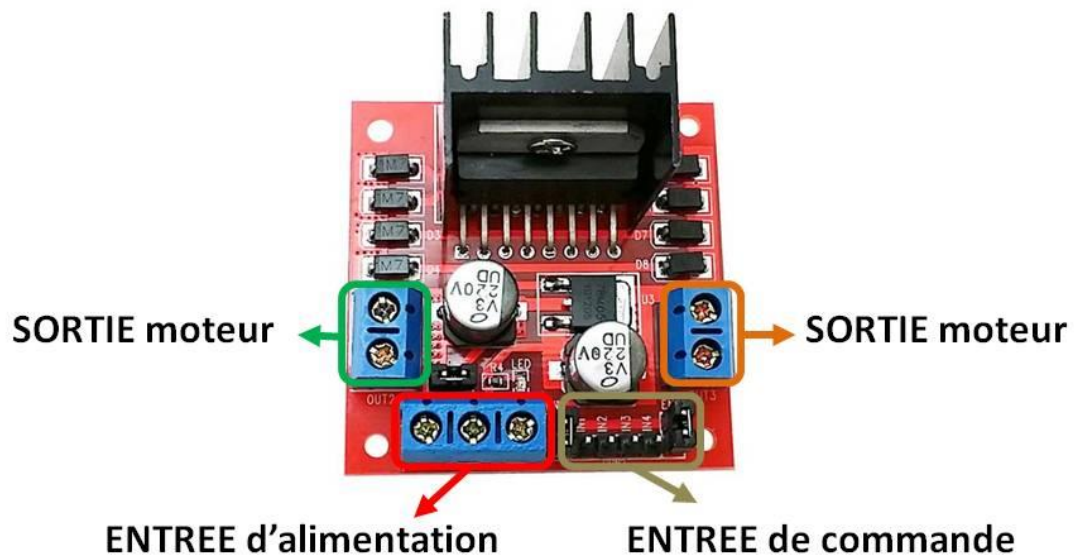


Figure 7 : Schéma module commande moteur

Cette carte électronique permet de piloter électroniquement le moteur. La carte ARDUINO seule ne suffit pas à la rotation de ce dernier. Un pont en H (montage électronique), comme intégré dans le module L298N, est nécessaire.

Moteur NEMA 17



Figure 8 : Moteur NEMA 17

Ce moteur transforme l'énergie électrique en énergie mécanique pour mouvoir notre système. Le moteur NEMA 17 est un moteur pas à pas. Celui-ci nous permet de contrôler avec précision l'angle qui lui est donné.

Bouton ON/OFF

Grâce à ce bouton il est possible, d'activer ou d'éteindre le système d'alimentation autonome. Ce dispositif est associé à une LED rouge. L'état du système d'alimentation lors du démarrage de

l'extrudeuse est normalement éteint. Par appui sur le bouton ON/OFF le système d'alimentation devient opérationnel. Dans ce cas la LED de signalisation s'allume pour confirmer que le système fonctionne.

LED alimenté en 5V en série avec une résistance de 220 Ω

Interrupteur alimenté en 5V en série avec une résistance de 10 k Ω

Module photorésistance et LED

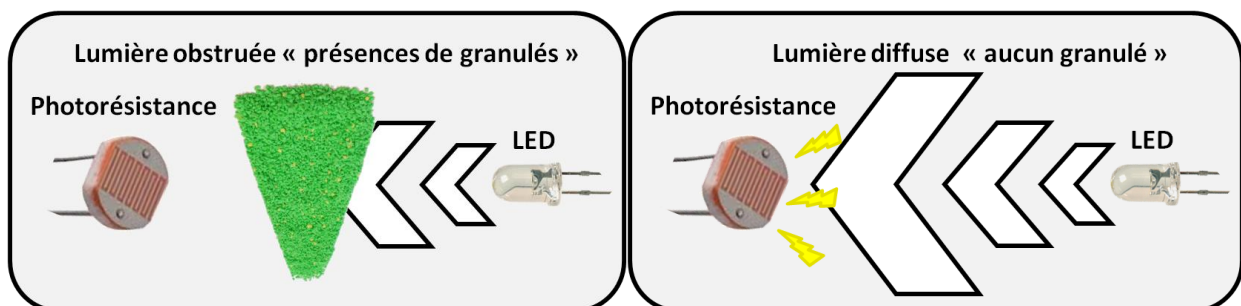
Ces deux modules déportés se décomposent en deux parties :

- Un module LED blanche diffusant de la lumière.
- Un module photorésistance permettant de capter la lumière diffusée par la LED.

Suivant s'il y a présence de granulés entre la LED et la photorésistance ou non, la photorésistance ne reçoit pas la même quantité de lumière et donc sa valeur varie. En fonction de cette valeur reçue par la carte ARDUINO, le système de distribution du granule se met en marche.

Schémas de principe

Les deux modules sont mis face à face



Ancien cadre expérimental :

Base d'essai :

- Réservoir rempli en granulés jusqu'à la LED verte
- Flèche moteur positionnée sur l'axe 0 du cadran angulaire
- LED éclaire pleinement la photorésistance et se positionne en face de celle-ci.

Limite de fonctionnement :

- Réservoir vide
- Plage d'angle moteur pour ouverture du système entre 0 et -110°



Figure 10 : Système fil lin avec anneau

Fixation avec anneau

Constatation : Perturbation dans la définition de l'angle d'origine après plusieurs essais

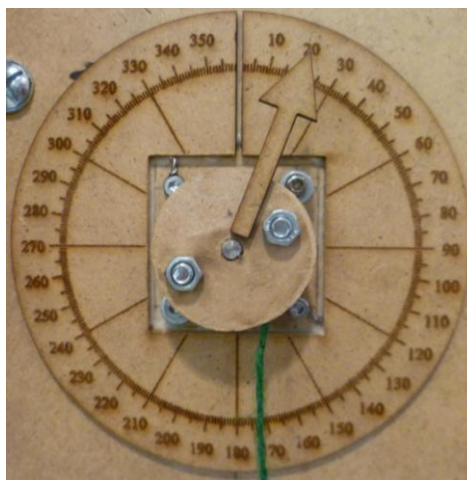


Figure 11 : Décalage cadran angulaire

Cause potentielle :

- 1) Le câble et/ou anneau se coince lors de la chute de granules dans l'entonnoir.
- 2) Problème dans le code lors de la conversion de l'angle d'ouverture en nombre de pas.
- 3) Problème technique provenant du moteur

Résultat après modification :

- 1) Modification de l'accroche (anneau retiré). Après modification le problème était moins fréquent mais s'est reproduit une fois.



Figure 12 : Système fil lin sans anneau

- 2) Fonction convertissant l'angle en pas. Nous saisissons directement le pas sans fonction de conversion. A l'issue du test le problème persistait. Donc le code n'était pas en cause.
- 3) Nous avons fais une batterie de tests de réglage d'angle sur le moteur sans l'avoir relié au dispositif. Le moteur fonctionne parfaitement.

Proposition de solution :

- Améliorer le prototype en remplaçant le fil de lin en fil de nylon
- Réduire l'ouverture de l'image « système sans anneau » pour éviter que les granules ne se coincent dedans.

Nouveau cadre expérimental :

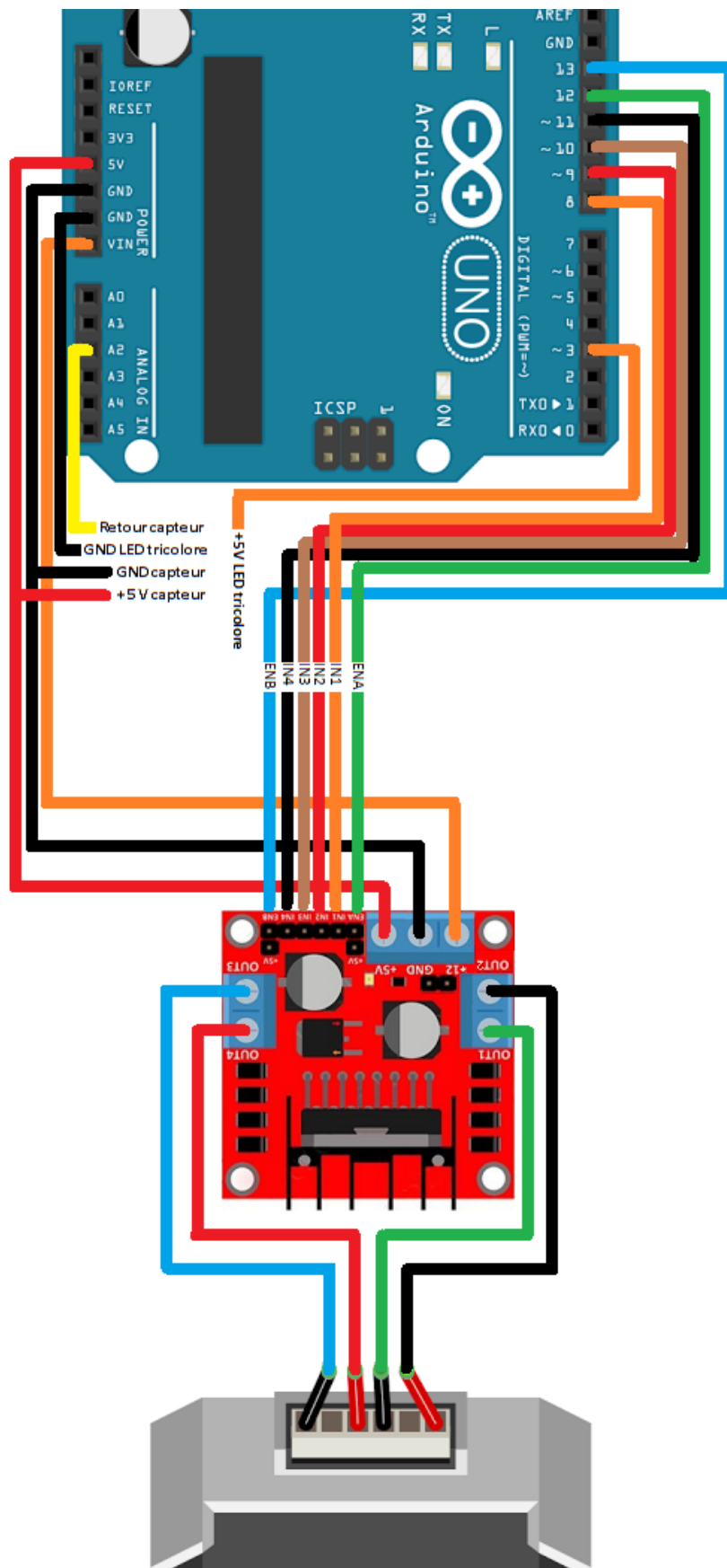
Base d'essai :

- Réservoir rempli en granulés jusqu'à la LED verte
- Flèche moteur positionnée sur l'axe 0 du cadran angulaire
- LED éclaire pleinement la photorésistance et se positionne en face de celle-ci.

Limite de fonctionnement :

- Réservoir vide
- Plage d'angle moteur pour ouverture du système entre 0 et -150°

4) Schéma de câblage boîtier électronique



Cas d'étude

Le seuil d'activation du système est défini en fonction de la valeur reçue par la photorésistance et de la puissance lumineuse émise par la LED :

LED alimentée en 5V en série avec une résistance de 220 Ω

Photo résistance alimentée en 5V en série avec une résistance de 10 k Ω

Valeur de fonctionnement :

Les valeurs ci-dessous ont été récupérées sur le moniteur série du logiciel ARDUINO

Cas photorésistance complètement obstruée (noir complet) = 650

Cas photorésistance directement éclairée par la LED = 850

Nous avons donc décidé de mettre notre seuil à 800 dans le programme ARDUINO.

Préparation avant allumage

Essai et calibration :

Essai avec une ouverture de ZZ et une durée de TT:

Granulés utilisés :

- Granule noir/blanc (ABS non recyclé)
- Granule blanc/bleu (PLA recyclé)

L'ouverture étant de ZZ le granule ne s'écoule pas durant toute la durée d'ouverture TT. Il laisse passer uniquement les granules les plus petits.

On constate donc un blocage au niveau du bouchon de la sortie de la trémie.

Nous avons également constaté un blocage au dessus de l'ouverture comme sur le schéma ci dessous lors de l'essai avec le PLA recyclé.

Nous avons également constaté un problème au niveau de la poulie moteur.

Modifier la valeur d'angle dans le programme :

Ouvrez le programme nommé « Programme_ARDUINO_FINAL » :

```
#define LED 5 // DEFINIR LE NUMERO DE PIN POUR LA LED
#define VIN 6 // DEFINIR LE NUMERO DE PIN POUR VIN (SERVANT A ALIMENTER LA LED ON/OFF)
#define BUTTON 2 //DEFINIR LA PIN D'ENTREE OU LE BOUTTON EST CONECTE

#include "Stepper.h"// INCLURE LA PROCEDURE STEPPER DANS LE CODE POUR DECLARER LE MOTEUR

int ENA=12; //Connecté sur votre Arduino, Pin 12
int IN1=8; //Connecté sur votre Arduino, Pin 8
int IN2=9; //Connecté sur votre Arduino, Pin 9
int ENB=13; //Connecté sur votre Arduino, Pin 13
int IN3=10; //Connecté sur votre Arduino, Pin 10
int IN4=11; //Connecté sur votre Arduino, Pin 11
int VLED=3; //Connecté sur votre Arduino, Pin 3

int val = 0; // VAL EST UNE VARIABLE UTILISEE POUR STOCKER LA VALEUR RETOURNE PAR LA PIN ON/OFF
int old_val = 0; // OLD_VAL STOCK LA VALEUR PRECEDENTE DE VAL POUR VERIFIER L'ETAT ACTUEL
int state = 0; // 0 = LED OFF ET 1 = LED ON
int var = 0;
int valeur2=0; // valeur2 contient la valeur retourné par le capteur photorésistance
int SeuilBas=700; // SeuilBas contient également la valeur retourné par le capteur photorésistance
int Angle = 110;
//pour un moteur de 200 pas par tour et brancher sur les broches 2, 4, 7, 8
Stepper moteur(200, 8, 9, 10, 11);
```

Valeur à modifier (exemple :
remplacer la valeur 110 par 140)

Pour modifier la valeur d'angle d'ouverture de notre système il n'y a qu'à modifier la valeur nommé « angle » dans le programme ARDUINO (encadré en vert ci-dessus).

Une fois la valeur modifier la valeur angle sera gérer dans le code automatiquement comme vous pouvez le voir ci-dessous :

```
if (state == 1)
{
  digitalWrite(LED, HIGH); // METTRE LED ON/OFF EN ON

  //----LECTURE VALEUR SEUIL BAS (si SB<700 = OBJET sinon PAS OBJET)
  valeur2 = analogRead(SeuilBas); // LECTURE VALEUR CAPTEUR PHOTORESISTANCE
  Serial.println(valeur2); //RECUPERATION DE LA VALEUR LU DANS LA VARIABLE VALEUR 2
  delay(10);

  if (valeur2 > 800)
  {
    moteur.step(-(Angle*200)/360); //choisir le nombre de pas à faire (Moteur à 200 pas)
    var = 0;
    while(var < 200){
      val = digitalRead(BUTTON); // LIRE LA VALEUR D'ENTREE BUTTON ET LA STOCKER DANS VAL
      if ((val == HIGH) && (old_val == LOW))
      {
        state = 1 - state; // SI C'EST VRAI ALORS MISE A 1 DE STATE POUR ALLUMER LED ON/OFF
        delay(10); // PAUSE
      }

      old_val = val; // SAUVEGARDER L'ETAT PRECEDENT DE VAL DANS OLD_VAL
      delay(10);
      var++;
    }
    moteur.step((Angle*200)/360); //choisir le nombre de pas à faire (Moteur à 200 pas)
    delay(10); //délai d'attente entre deux mouvements
  }
}
```