Traitement d'une eau trouble (mesure de turbidité)

Informations

- Mohammed EL Amine CHOUAF(mohammed_el_amine.chouaf@etu.sorbonne-universite.fr)
- Loréna COPPET (lorena.coppet@etu.sorbonne-universite.fr)
- Sara Yasmine GAOUA (sarra_yasmine.gaoua@etu.sorbonne-universite.fr)
- Chaza TOUATI (chaza.touati@etu.sorbonne-universite.fr)

- Master 2 de Chimie parcours Ingénierie Chimique : MU5Ci - 803 Optimisation et contrôle des procédés
- Période: 11/12/2023 - 12/02/2024

Contexte

On peut définir la turbidité comme la teneur en matériaux en suspension dans un liquide. En d'autres termes, on va mesurer la clarté d'un liquide. La nature de la suspension peut être diverse et variée.

Dans notre cas, on souhaite traiter une eau trouble (synthétique) par filtration (sur une colonne contenant différentes couches filtrantes: gravier, sable fin et gros sable). Pour se faire on cherchera à mesurer en continu la turbidité du filtrat et ainsi instaurer une boucle de recyclage en fonction de la valeur obtenue et des critères définis et tout cela en automatisant le système (Arduino).

Ce projet cumul deux disciplines que sont la chimie et l'automatisme. Le fait d'automatiser la filtration nous permet d'obtenir un résultat répétable, facilitant la recherche et le développement du procédé.

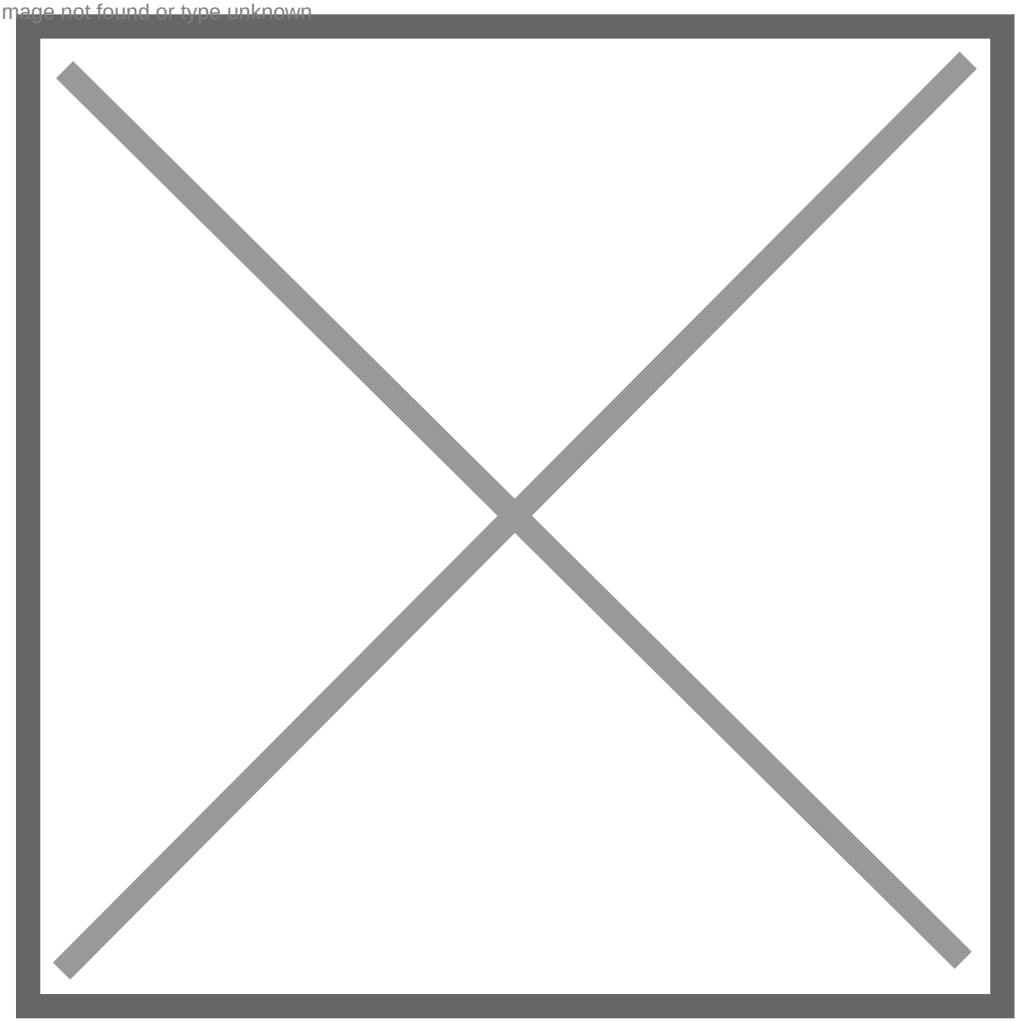
Mode opératoire

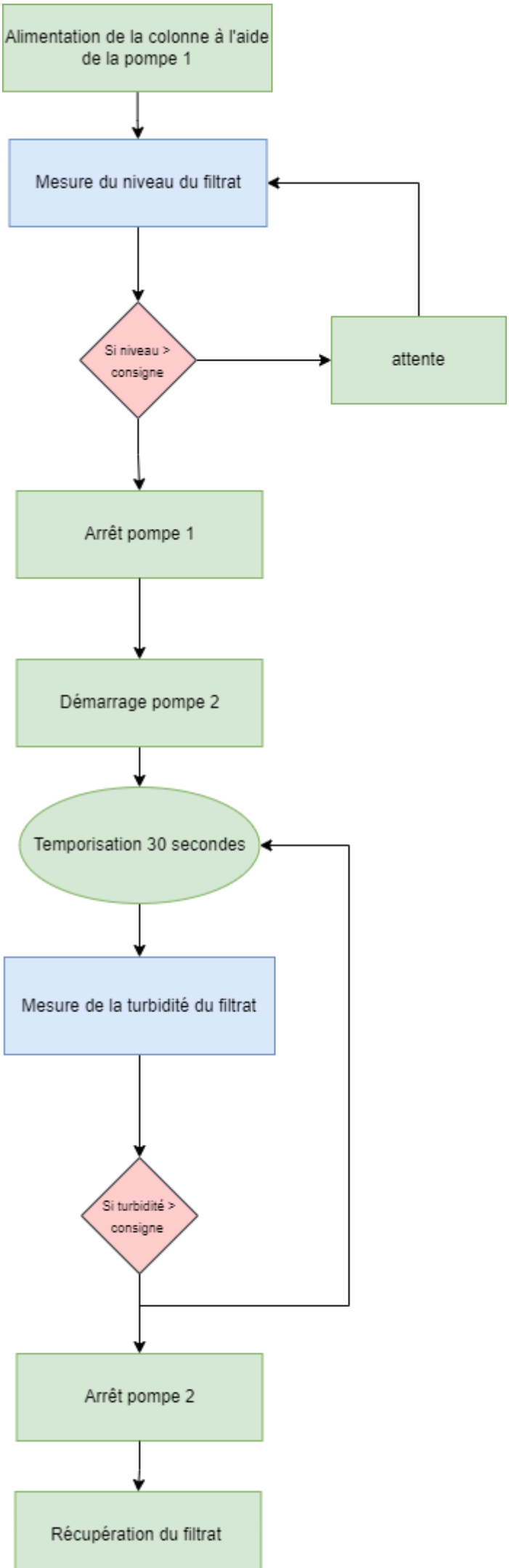
Objectif: Mesure en continu de la turbidité d’une eau trouble traitée par filtration.

- Alimentation de la colonne via une pompe électrique (pompe 1)
- Récupération du filtrat et mesure de la turbidité à l’aide d’une sonde (capteur de turbidité)

- Tant que la consigne n’est pas atteinte, recyclage du filtrat dans la colonne à l’aide d’une pompe (pompe 2)
- Contrôle du niveau d’eau dans la colonne et celui de la récupération du filtrat avec des capteurs de niveau

Image not found or type unknown




Alimentation de la colonne à l'aide de la pompe 1
Mesure du niveau du filtrat
Si niveau > consigne
attente
Arrêt pompe 1
Démarrage pompe 2
Temporisation 30 secondes
Mesure de la turbidité du filtrat
Si turbidité > consigne
Arrêt pompe 2
Récupération du filtrat

Composants

- 2 pompes électriques
- 2 réservoirs
- 2 capteurs de niveau
- Une colonne de filtration (conçue à l'imprimante 3D)
- Filtres
- 1 capteur de turbidité
- Eau trouble synthétique
- 1 carte Arduino
- Branchements
- 2 alimentations 12 V
- 2 moteurs
- 2 agitateurs (conçus à l'imprimante 3D)
- Piles de 3 V

Construction

Étape 1

Plan du procédé

Listing du matériel

Étape 2

Construction de la maquette

Recherche du code

Connexion avec Arduino

Étape 3

Éssais

Journal de bord

16/10/2023

Choix du projet et du groupe de travail

27/10/2023

Réunion de projet: définition des enjeux

04/11/2023

2ème réunion de projet: Réalisation du schéma du procédé et listing du matériel

08/11/2023

Entretien avec le tuteur afin d'affiner nos idées; discussion sur la faisabilité des réalisations; listing du matériel disponible

10/11/2023

Ajustement et correction du schéma suite à l'entretien

21/11/2023

Récupération du matériel à Sorbonne Université - plateforme d'Ingénierie Chimique

23/11/2023

Rangement du matériel au Fablab dans une boite prévu à cet effet

05/12/2023

Récupération de gravier et sable (fin, moyen) nécessaire à la filtration

13/12/2023

Récupération du support en bois pour le montage du procédé

21/12/2023

1ère rencontre au fablab pour observer et découvrir le matériel disponible

22/12/2023

Montage du support en bois

28/12/2023

Code préliminaire: recherche du code Arduino pour chaque composant de notre procédé

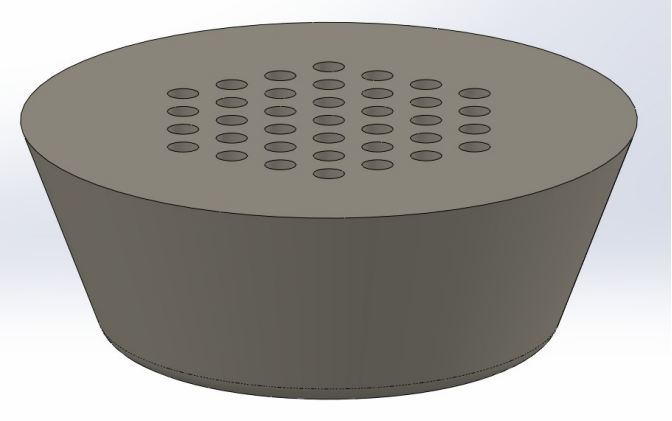
08/01/2024

Impression de la colonne de filtration avec l'imprimante 3D disponible au Fablab

Dimensions : il s'agit d'un entonnoir avec un grand diamètre externe de 14 cm, un grand diamètre interne de 13,5 cm, un petit diamètre externe de 2 cm, un petit diamètre interne de 1,3 cm, une hauteur totale de 21 cm, une hauteur du cône de 16 cm et une hauteur du tube de 5 cm

09/01/2024

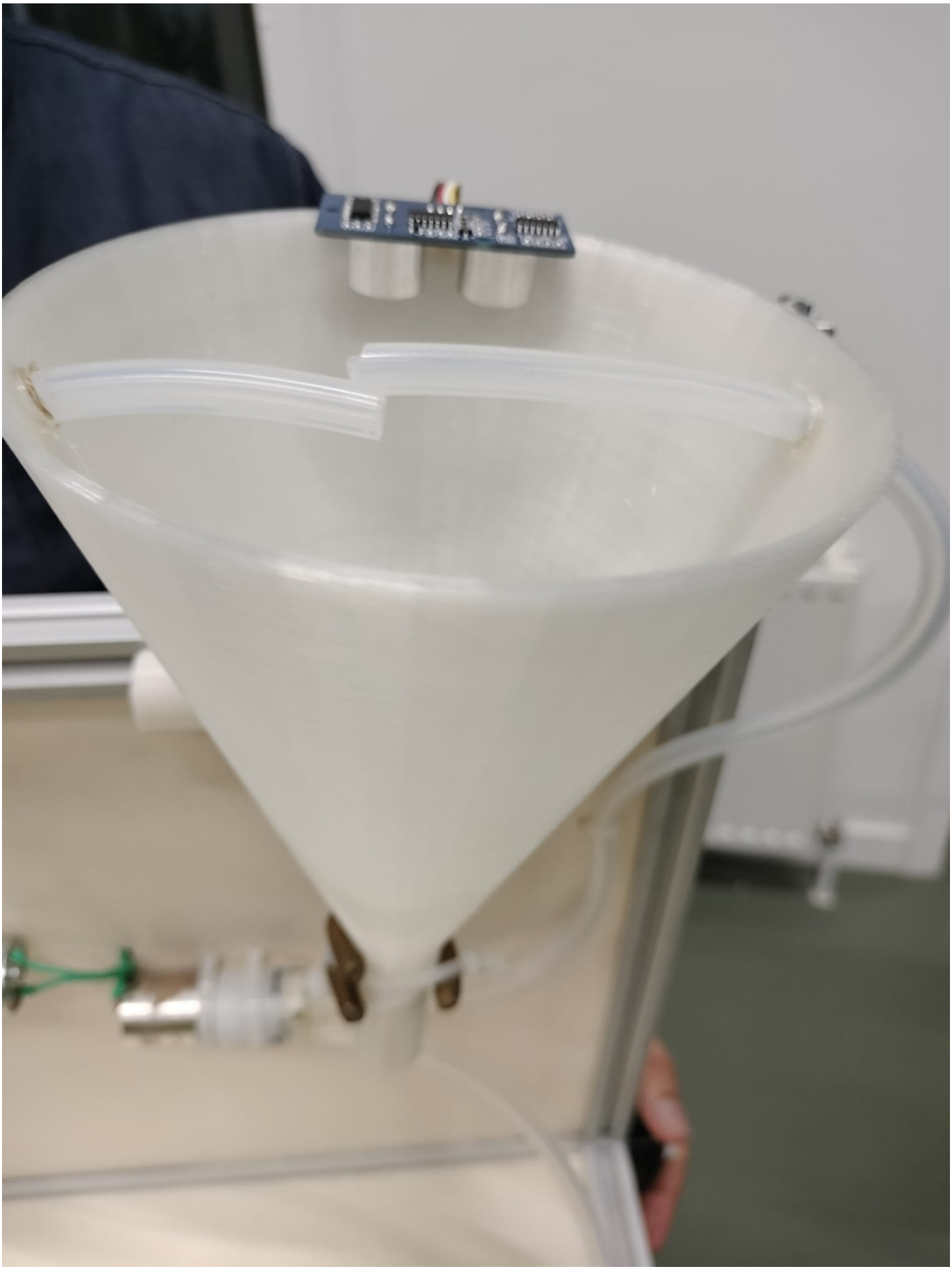
Dimensionnement du support du filtre

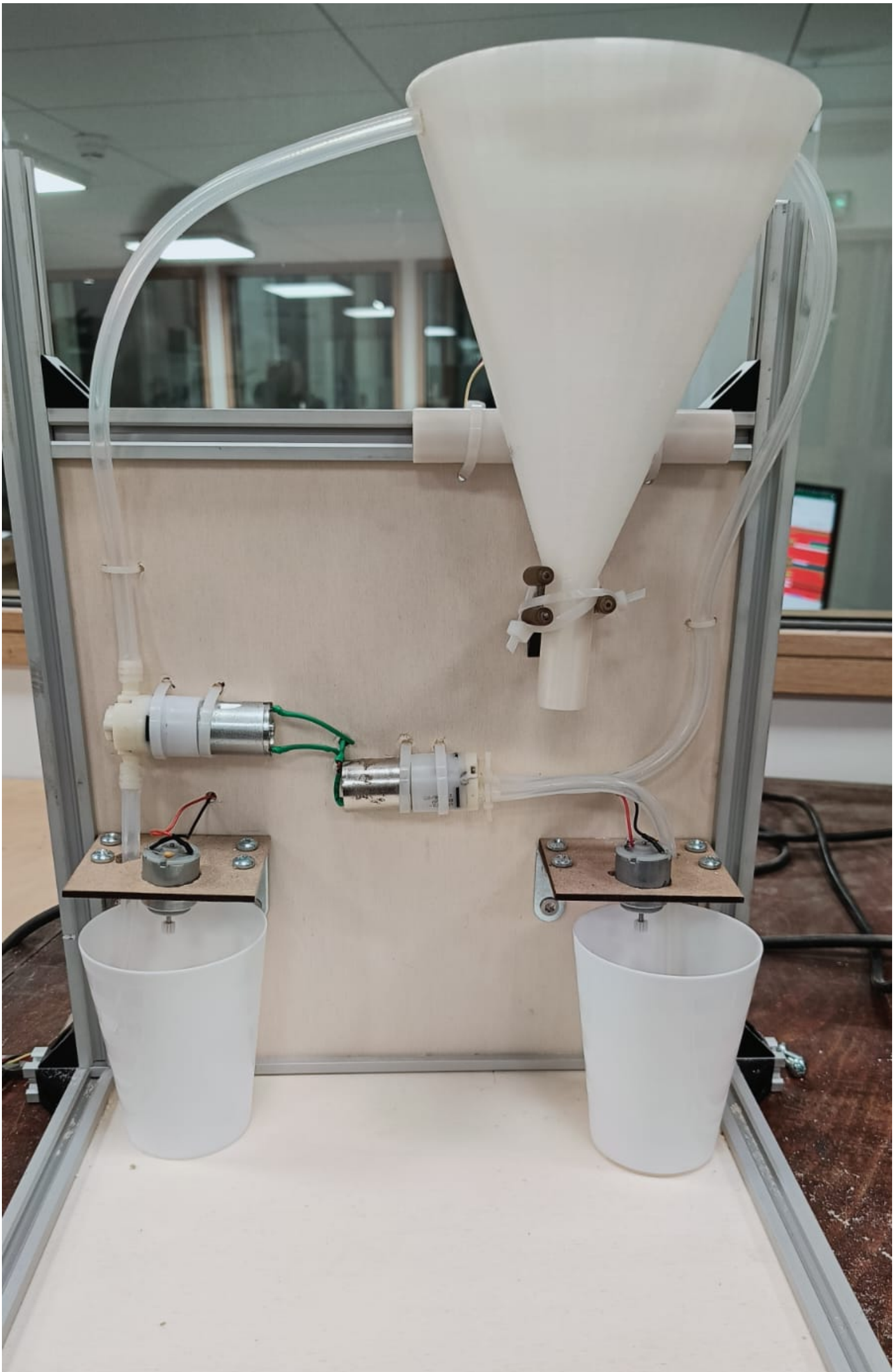


19/01/2024

Fixation de la colonne sur le support en bois

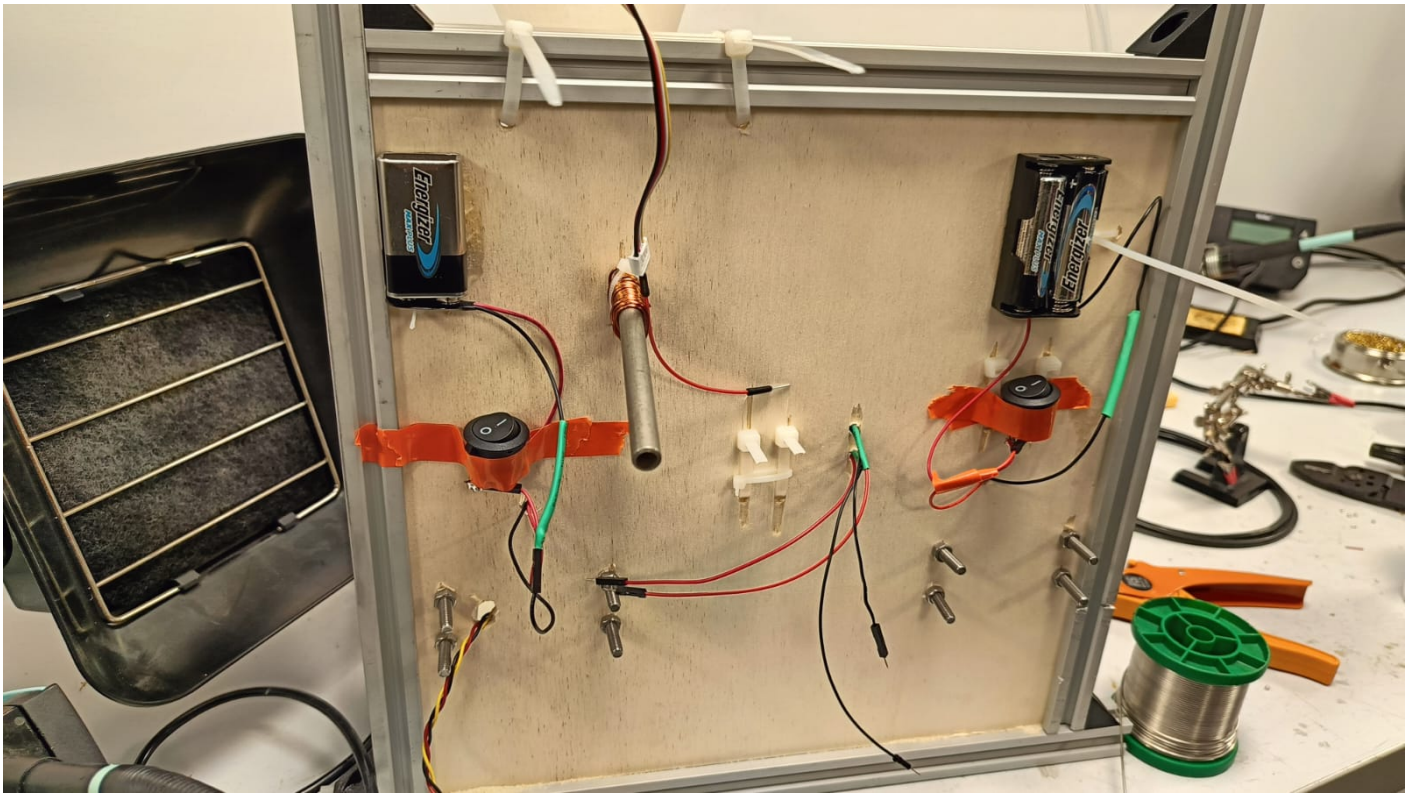
22/01/2024

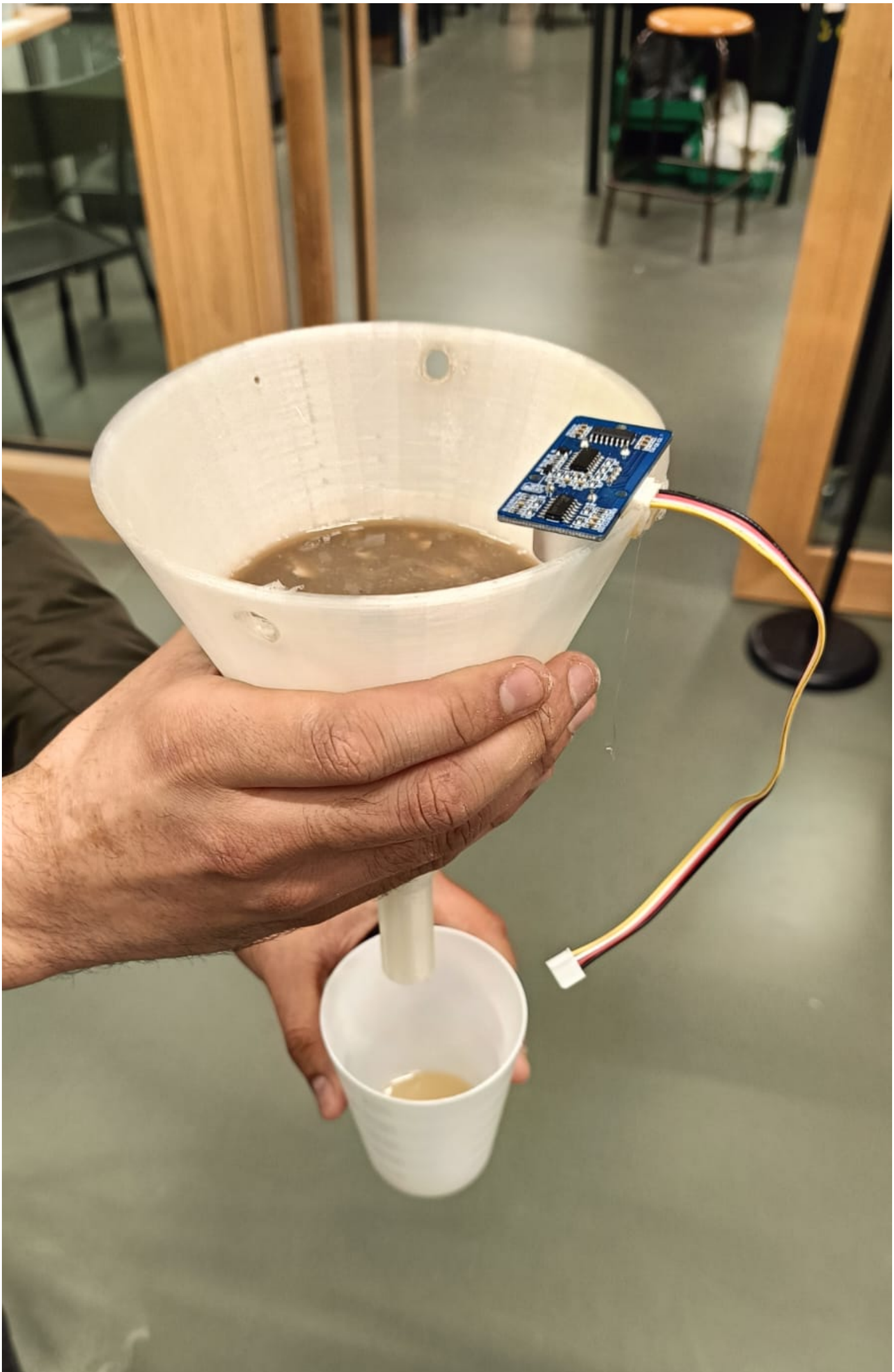


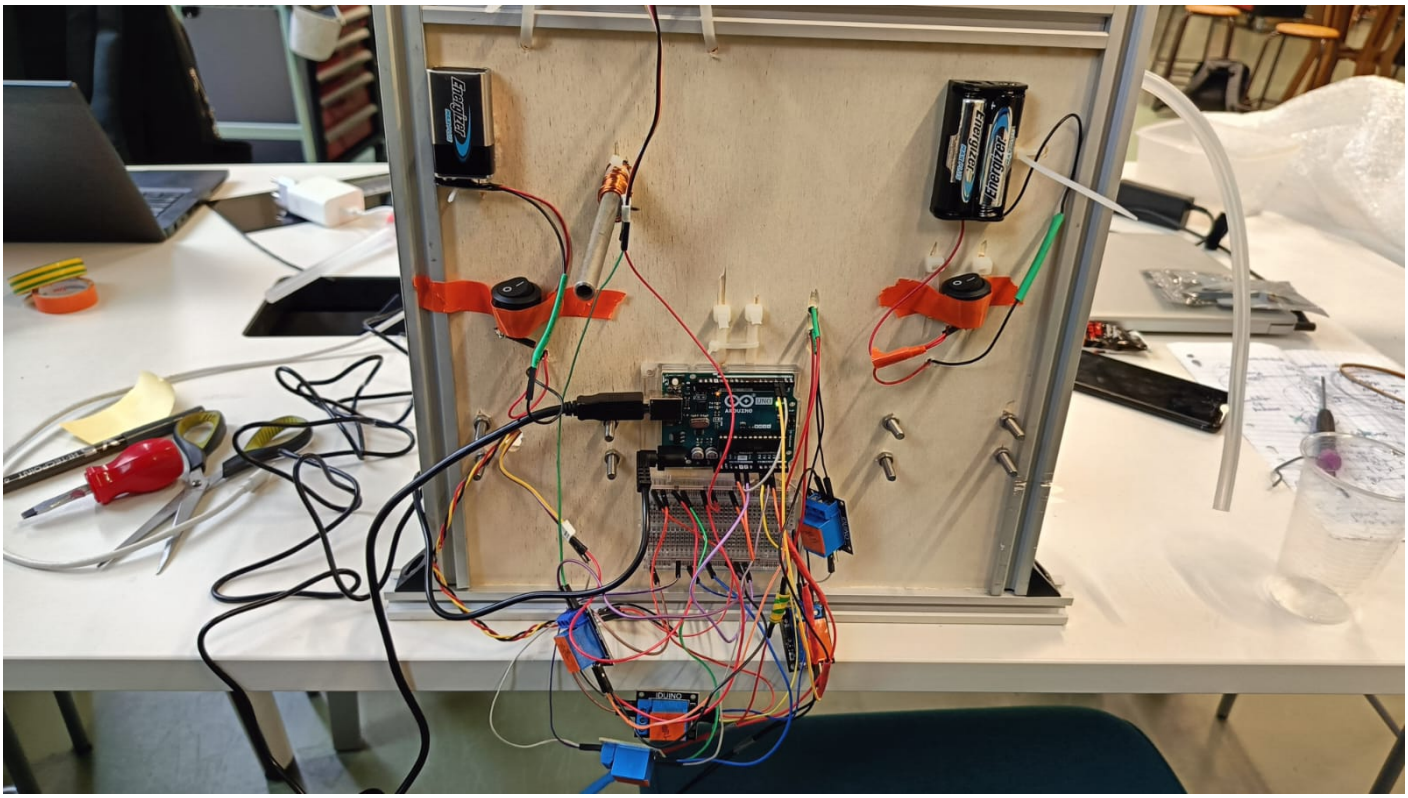


Impression 3D des mobiles d'agitation

Dimensions : longueur arbre d'agitation 9 cm, diamètre hélice 3 cm


Energizer
Energizer



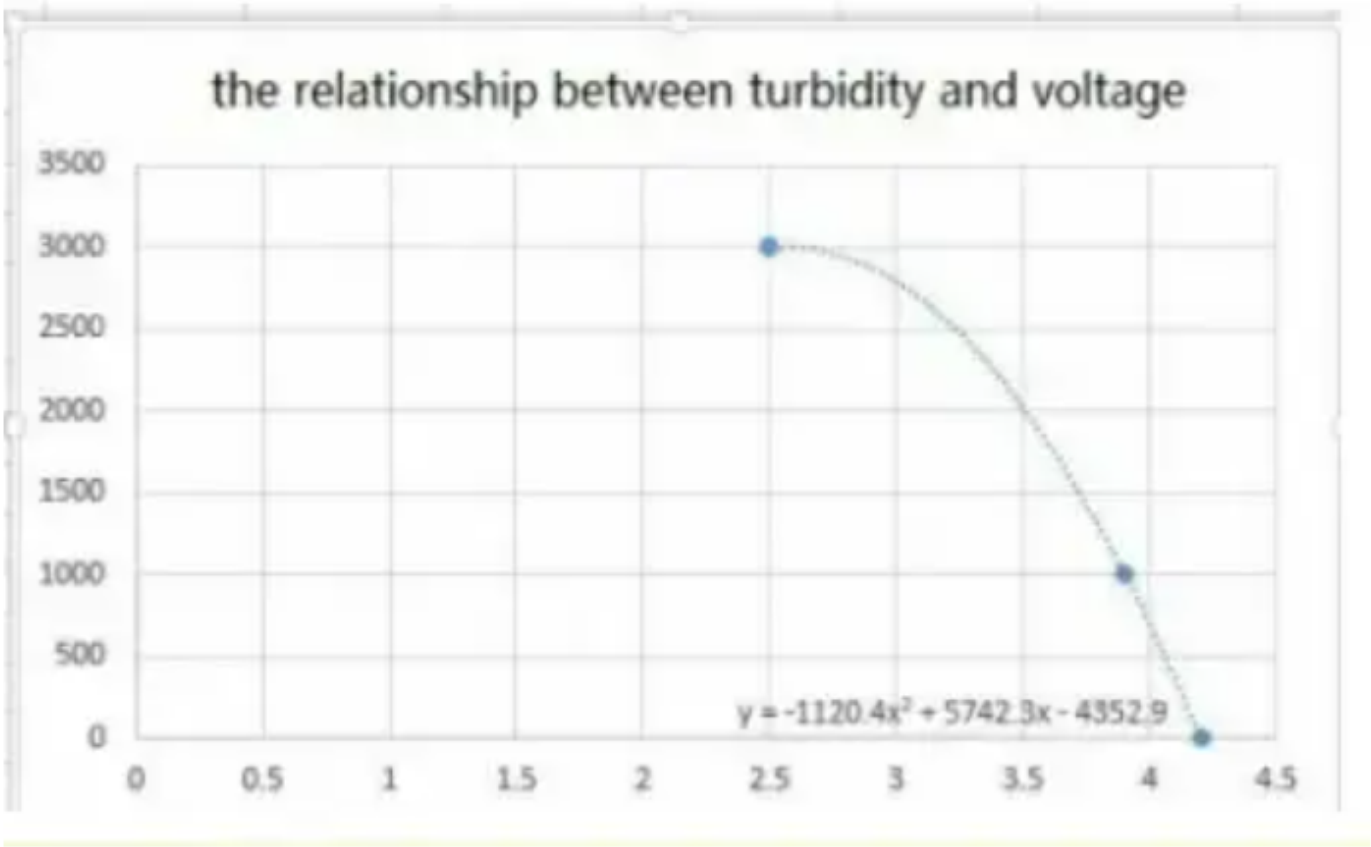


02/02/2024

Réalisation des essais de fonctionnement. Nous avons soulevé plusieurs points bloquants :
- Absence de connecteur pour le capteur de turbidité
- Défaillance du capteur de niveau de la colonne
- Agitation du filtrat trop importante : il faudrait pouvoir piloter la fréquence de rotation du moteur à l'aide d'un potentiomètre

04/02/2024

Recherche du manuel du capteur de turbidité. Informations sur la relation entre le voltage et la turbidité du capteur (modèle sen0189)



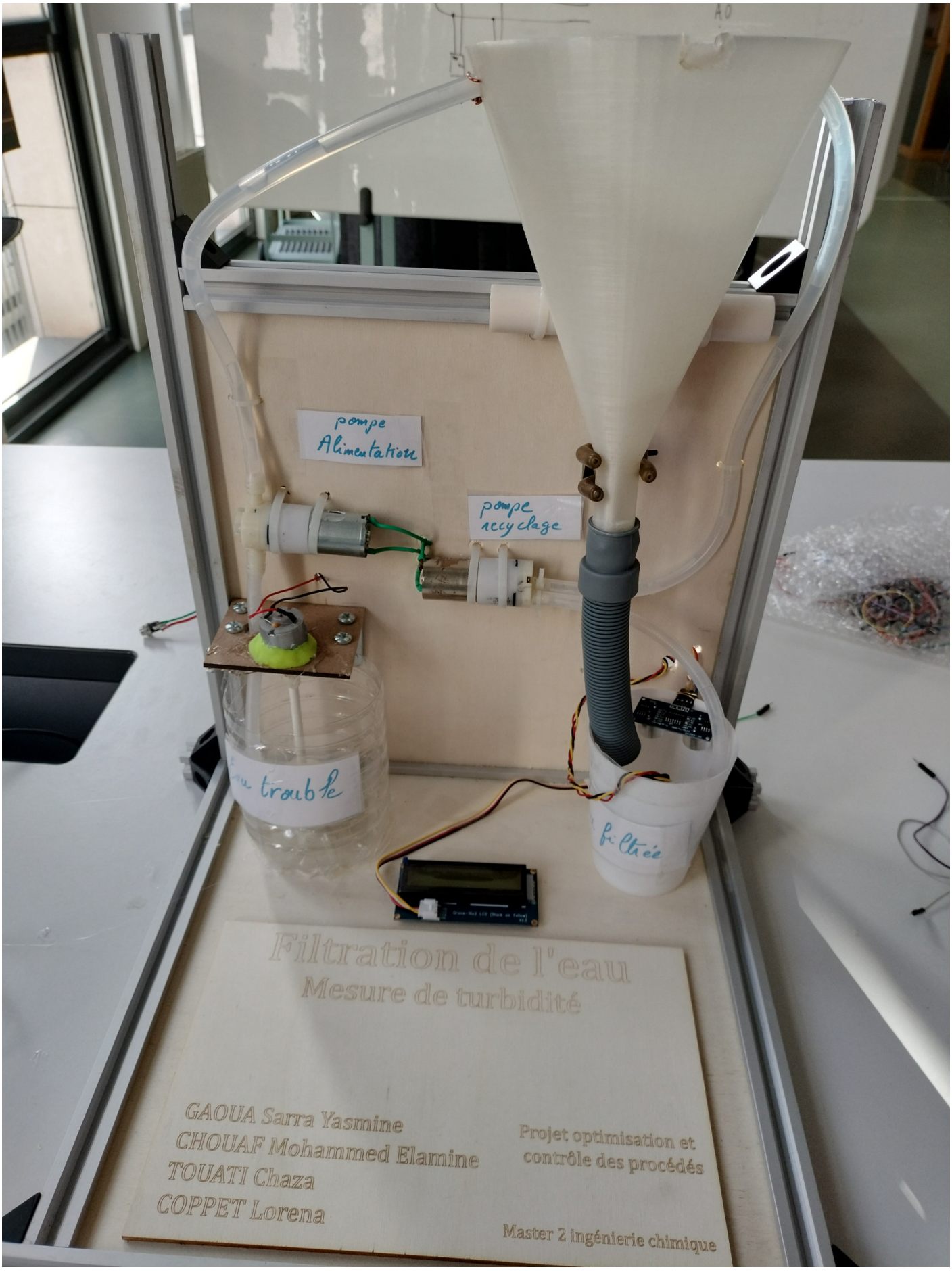
Cette courbe est la fonction de transfert de la tension en turbidité.

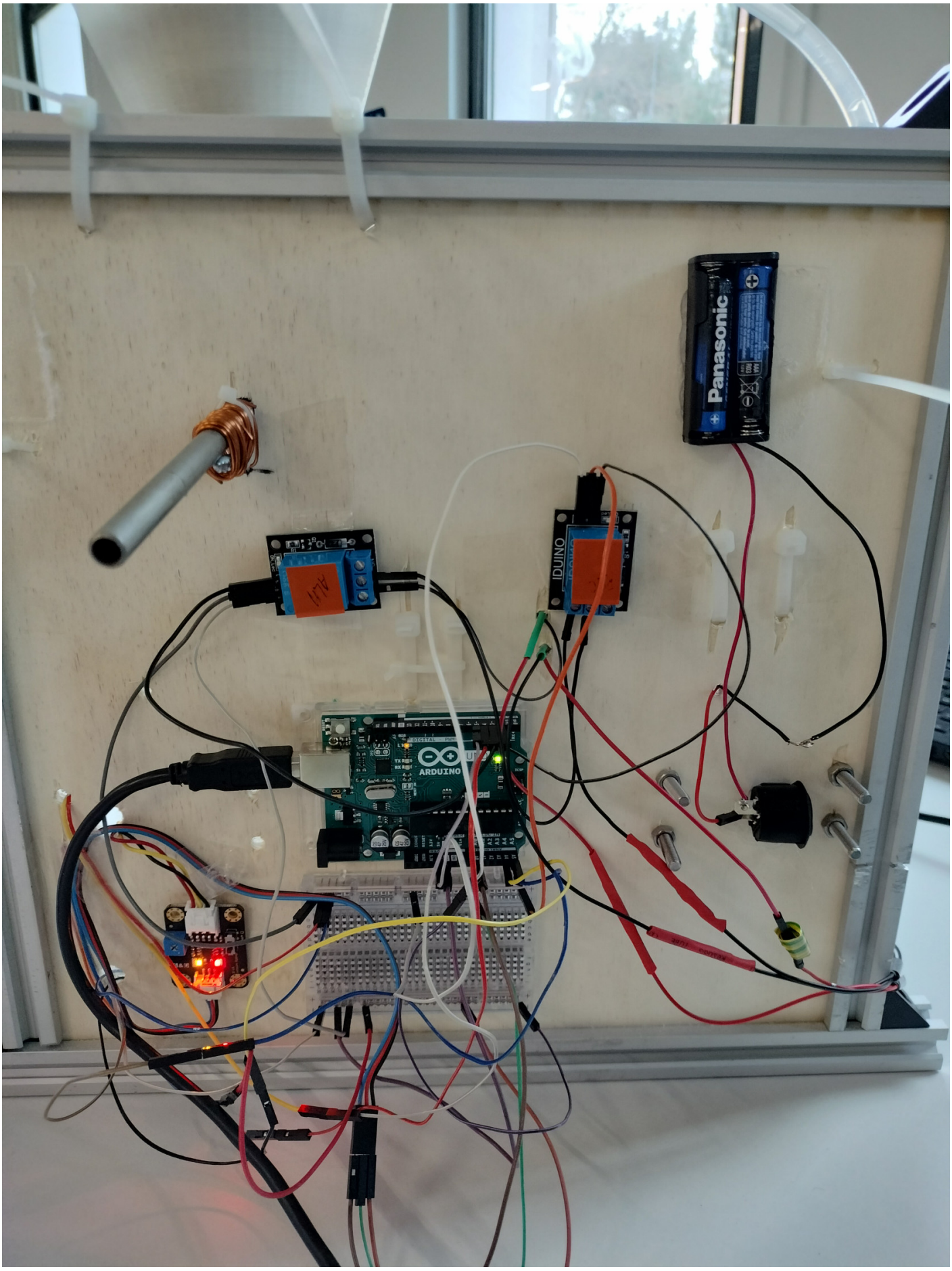
Mesure de la turbidité de la solution d'alimentation : 3000 ntu (eau turbide)

Détermination de la consigne : 1763 ntu (eau claire)

09/02/2024

Dernier montage


pompe
Alimentation
pompe
recyclage
trouble
filtrée
Filtration de l'eau
Mesure de turbidité
GAOUA Sarra Yasmine
CHOUAF Mohammed Elamine
TOUATI Chaza
COPPET Lorena
Projet optimisation et
contrôle des procédés
Master 2 ingénierie chimique



Pour le montage final, nous avons gardé l'agitation seulement au niveau de l'alimentation et pas au niveau du bac de filtrat pour assurer une bonne mesure de niveau et de turbidité.

Code

```
#include &lt;Wire.h&gt;
#include &lt;rgb_lcd.h&gt;

const int relaisPompe1 = 3; // Broche de commande pour la pompe d&#39;alimentation
const int relaisRecyclage2 = 2; // Broche de commande pour la pompe de recyclage 2
const int trigPin = 5; // Broche de déclenchement pour capteur de niveau
const int echoPin = 4; // Broche de réception pour capteur de niveau
const int sensorPin = A0; // Broche pour le capteur de turbidité
const int d1 = 2; // Seuil de distance pour le capteur de niveau
const float TS = 1703; // Turbidité souhaitée

void activerPompe(int relais, unsigned long duree);
void activerCapteurNiveau();
int mesurerDistance();
float mesurerTurbidite();

void setup() {
// Initialisation des broches des relais en sortie
pinMode(relaisPompe1, OUTPUT);
pinMode(relaisRecyclage2, OUTPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(sensorPin, INPUT);

Serial.begin(9600); // Initialisation de la communication série
}

void loop() {
// Déclencher la pompe 1 pendant 5 secondes
activerPompe(relaisPompe1, 5000);

// Attendre 10 secondes tout en mesurant la distance avec le capteur de niveau
unsigned long startTime = millis();
while (millis() - startTime &lt; 10000) {
activerCapteurNiveau();
int distance = mesurerDistance();
if (distance == d1) {
digitalWrite(relaisPompe1, LOW); // Arrêter la pompe si la distance atteint d1
```

```
float ntu = mesurerTurbidite();

// Si la turbidité est supérieure à la turbidité souhaitée, activer la pompe de recyclage 2 pendant
10 secondes
if (ntu &gt; TS) {
activerPompe(relaisRecyclage2, 10000);
}

// Vérifier si la turbidité a atteint la valeur souhaitée (TS) pour arrêter le processus
if (ntu == TS) {
// Arrêter le processus
return;
}
}
delay(500); // Attendre un court laps de temps avant de refaire une mesure
}

// Attendre 5 minutes avant de répéter le processus
delay(5 * 60 * 1000); // 5 minutes en millisecondes
}

// Fonction pour activer une pompe pendant une durée spécifiée
void activerPompe(int relais, unsigned long duree) {
digitalWrite(relais, HIGH);
delay(duree);
digitalWrite(relais, LOW);
}

// Fonction pour activer le capteur de niveau
void activerCapteurNiveau() {
// Envoyer une impulsion ultrasonique
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
}

// Fonction pour mesurer la distance avec le capteur de niveau
int mesurerDistance() {
```

```
// Mesurer la durée du signal écho
long duration = pulseIn(echoPin, HIGH);

// Calculer la distance en centimètres
int distance = duration * 0.034 / 2;

// Afficher la distance mesurée
Serial.print(&quot;Distance: &quot;);
Serial.print(distance);
Serial.println(&quot; cm&quot;);

return distance;
}

// Fonction pour mesurer la turbidité
float mesurerTurbidite() {
float volt = 0;
for (int i = 0; i &lt; 800; i++) {
volt += ((float)analogRead(sensorPin) / 1023) * 5;
}
volt = volt / 800;
volt = round(volt * 100) / 100; // Arrondir à deux décimales
float ntu;
if (volt &lt; 1) {
ntu = 3000;
} else {
ntu = -1120.4 * volt * volt + 5742.3 * volt - 4353.8; // Utilisation de l&#39;opérateur * pour élever au
carré
}
delay(10);
return ntu;
}
```

Revision #10
Created 28 December 2023 18:36:18 by Touati Chaza
Updated 19 February 2024 09:56:46 by Coppet Lorena