

# Projet : Régulation de conductivité dans un réacteur

## UE MU5CI823- Optimisation, contrôle et digitalisation des procédés

### Groupe 1

#### Nom et prénom des membres de l'équipe :

BEN SALHA Firas

KUTA Sara

LAICHE Ines

BAHEDA Abou Bekr

YU Jiawen

**Promotion :** Master 2 de Chimie spécialité ingénierie chimique 2025-2026

**Date de début du projet :** Octobre 2025



**Date de fin du projet :** Estimation fin janvier

#### Introduction :

Ce projet s'intéresse à la régulation de la conductivité dans un réacteur à l'aide d'un système de contrôle automatisé. L'objectif est de mesurer en continu la conductivité de la solution, de la comparer à une valeur de consigne et d'agir sur un actionneur (pompe ou électrovanne) afin de maintenir le procédé dans les conditions souhaitées. Ce travail permettra de mettre en œuvre une boucle de régulation en temps réel et d'illustrer les principes fondamentaux du contrôle et de l'optimisation des procédés étudiés dans l'UE MU5CI823.

## Matériels :

- 3 cuves - pour l'eau dure, pour l'eau claire, pour le mélange (le réacteur) une boîte de tomate concentré
- 2 tubes PVC pour connecter les cuves avec le réacteur
- 2 électrovannes
- Raccord rapides
- Support pour le montage (tournevis, vis)
- Sonde de conductivité + conductimètre
- Capteur de niveau + Alarme
- Transmetteur (l'écran dans le schéma)
- Arduino, carte mere, Relay , BREADBOARD
- Alimentation électrique (batterie 12 V)
- Câblage et bornier
- NaCl
- Eau distillée / déionisé
- Bécher, entonnoirs, balance, pipettes pour préparer les solutions
- Bac de récupération (sécurité)

## Avancement du projet :

### Compte rendu du 13 octobre 2025

13/10/2025 Les membres de l'équipe se sont réunis afin de définir le sujet du projet et d'échanger sur les différentes idées proposées. Quatre projets ont été suggérés et discutés collectivement, chacun présentant des approches et des enjeux techniques distincts. Cette réunion a permis à chaque membre du groupe d'exprimer ses préférences, d'évaluer la faisabilité des propositions et d'identifier les objectifs pédagogiques associés. À l'issue des échanges, un consensus s'est progressivement dégagé, posant ainsi les bases du choix final du projet et de l'organisation du travail à venir.

Au cours de la phase de réflexion initiale, quatre projets ont été envisagés par l'équipe, chacun illustrant un aspect différent de la régulation et du contrôle des procédés :

- **Projet 1 : Régulation de vitesse entre deux trains transportant des produits chimiques**

Ce projet visait à synchroniser la vitesse de deux convois afin d'assurer un transfert sécurisé ou coordonné de matières chimiques, mettant en jeu des problématiques de sécurité et de communication entre systèmes.

- **Projet 2 : Remplissage automatique des wagons et régulation de niveau**

L'objectif de ce projet était de concevoir un système permettant de gérer automatiquement le remplissage de wagons de transport, tout en garantissant le maintien du niveau dans une plage de sécurité définie.

- **Projet 3 : Régulation de température dans un ballon d'eau chaude sanitaire**

Ce projet consistait à développer une boucle de contrôle permettant de réguler la température d'un ballon ECS, en intégrant des éléments de chauffage, de mesure et éventuellement de sécurité thermique.

- **Projet 4 : Régulation de conductivité dans un réacteur**

Le dernier projet portait sur le contrôle de la conductivité au sein d'un réacteur, en ajustant la composition du milieu via un système automatisé. Ce sujet combine instrumentation, contrôle et compréhension physico-chimique du procédé.

### **Compte rendu du 15 octobre 2025 :**

Le 15 octobre 2025, les quatre projets identifiés ont été présentés à notre enseignant, M. Jérôme Pulpytel, afin d'obtenir son avis et sa validation. Après discussion, il a retenu le projet n°4 portant sur la régulation de la conductivité dans un réacteur, considérant sa pertinence pédagogique et sa faisabilité expérimentale. Il nous a également demandé d'établir une liste détaillée du matériel nécessaire à la mise en œuvre du dispositif, afin de planifier au mieux les étapes de réalisation du projet.

### **Compte rendu du 17 octobre 2025 :**

Le 17 octobre 2025, une réunion a été organisée sur Zoom afin de poursuivre l'avancement du projet et d'échanger plus précisément sur le matériel nécessaire à sa réalisation. Lors de cet échange, les membres du groupe ont discuté des capteurs, actionneurs et composants électroniques indispensables à la mise en place de la régulation de conductivité dans le réacteur. Les différentes options techniques ont été comparées afin de sélectionner des équipements adaptés aux objectifs du projet et compatibles avec les contraintes expérimentales. Cette réunion a permis d'établir une première liste détaillée du matériel et de répartir les prochaines tâches entre les membres de l'équipe. La commande du matériel nécessaire a ensuite été effectuée par M. Jérôme Pulpytel.

### **Compte rendu du 17 novembre 2025 :**

Le matériel commandé est arrivé et a été récupéré par le groupe 1, puis transporté au FabLab afin de procéder à l'inscription et d'obtenir l'autorisation d'y travailler. Par la suite, une réunion de 30 minutes a été organisée pour discuter de la disponibilité des membres de l'équipe et planifier les séances de travail en commun. Cette réunion a également permis de définir l'organisation générale pour la mise en route du projet et la bonne utilisation du matériel au FabLab.

### **Compte rendu du 19 novembre 2025 :**

Ce jour-là, nous avons travaillé sur la réalisation du support de notre projet. Nous avons commencé par découper deux planches en bois, chacune de dimensions 30 × 30 cm, à l'aide d'une scie. L'épaisseur a été choisie de manière à garantir une bonne stabilité du montage final. (6mm)

Nous avons ensuite préparé un support supplémentaire en réutilisant une ancienne planche de bois déjà trouée, afin d'éviter le gaspillage de matériel. Ce support a été fixé sur la structure en aluminium à l'aide d'équerres de fixation (équerres métalliques en forme de L), ce qui a permis d'assurer une bonne tenue mécanique.

Nous avons également réalisé deux ouvertures circulaires dans la planche supérieure à l'aide d'une scie-cloche, ces trous étant destinés à accueillir les réservoirs prévus pour le fonctionnement du système.

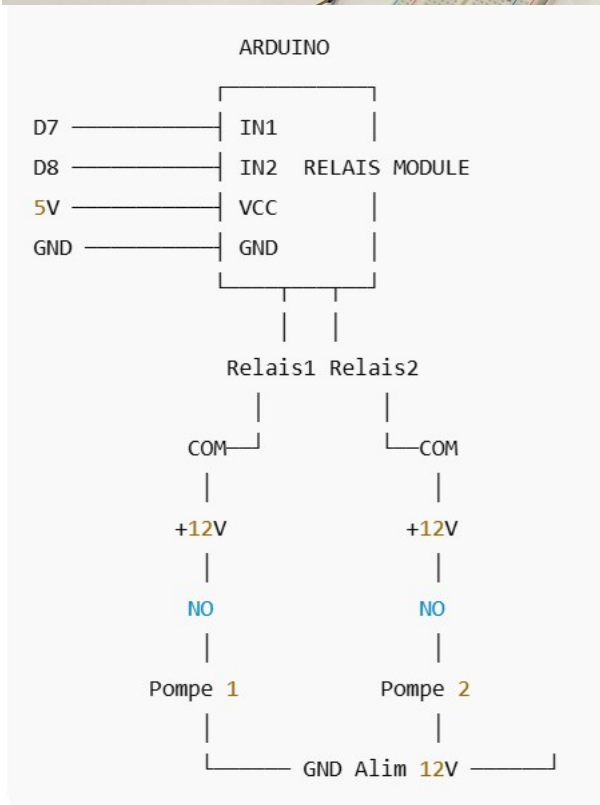
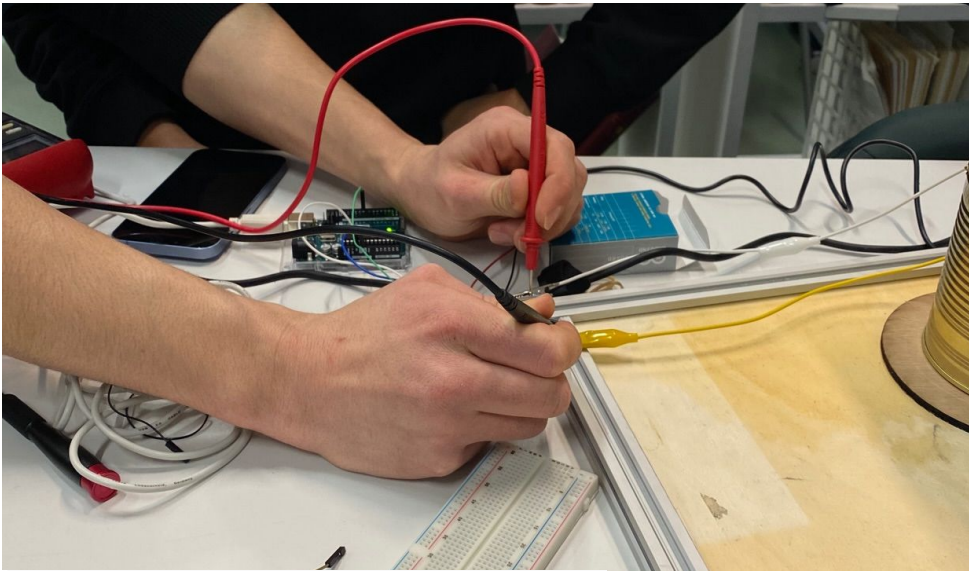
L'ensemble a ensuite été monté afin de vérifier l'ajustement, la stabilité et l'emplacement des réservoirs.

La prochaine intervention consistera à ramener les réservoirs et le réacteur, puis à procéder à la fixation des vannes et à leur installation sur la structure.



### **Compte rendu du 27 novembre 2025 :**

Ce jour-là, nous avons poursuivi le travail sur l'intégration des différents éléments du système sur le support. Nous avons d'abord procédé à l'installation des vannes sur la structure. Cependant, après réflexion sur la conception du montage et la simplicité du circuit, nous avons décidé de modifier le dispositif en retirant les vannes et en conservant uniquement les pompes pour assurer la circulation des fluides.



Ce schéma représente le câblage d'un système dans lequel un Arduino commande les deux pompes 12 V à l'aide d'un module relais à deux canaux. Les broches numériques D7 et D8 de l'Arduino sont reliées aux entrées IN1 et IN2 du module relais, tandis que le 5 V et le GND de l'Arduino alimentent le module. Chaque relais fonctionne comme un interrupteur commandé. La borne COM de chaque relais est connectée au +12 V de l'alimentation, et la borne NO (Normally Open) est reliée à une pompe. L'autre borne de chaque pompe est reliée au GND de l'alimentation 12 V. Lorsque l'Arduino active un relais, le contact se ferme entre COM et NO, ce qui alimente la pompe correspondante et la met en fonctionnement.

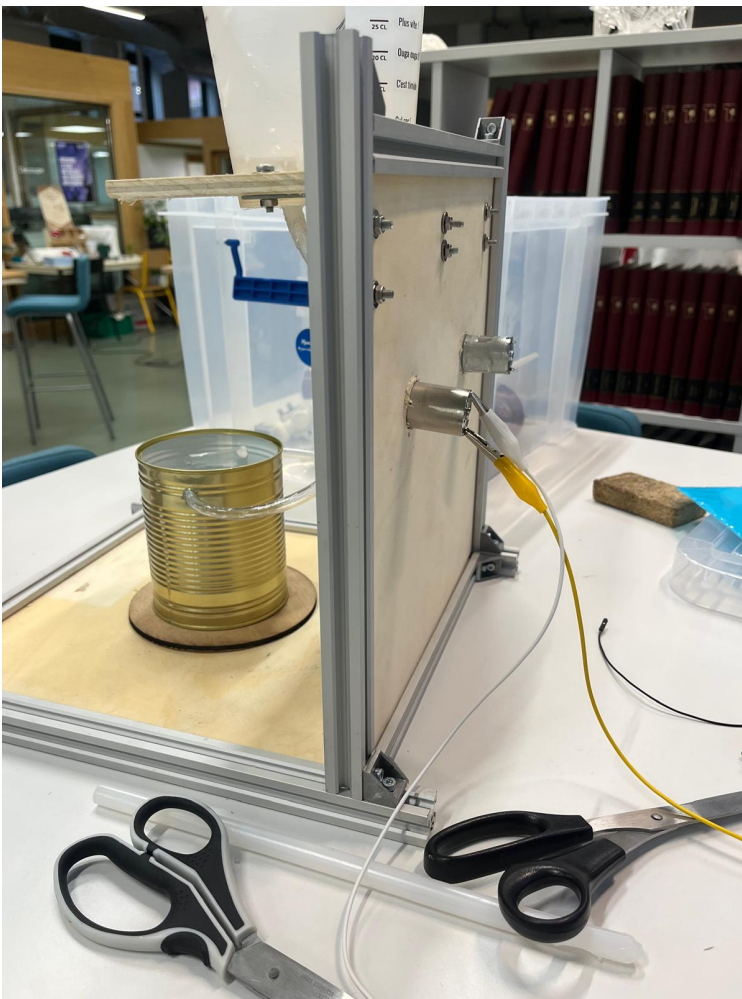
Nous avons ensuite continué à travailler sur le support afin d'améliorer la fixation et l'organisation des différents composants. Le réacteur a été installé sur la structure, puis les tuyaux ont été mis en place pour relier les différents éléments du système, en veillant à assurer un positionnement correct et une bonne tenue mécanique.

Enfin, nous avons réalisé des essais de fonctionnement des pompes à l'aide d'une carte Arduino. Ces tests ont permis de vérifier leur mise en marche, leur contrôle et leur bon fonctionnement dans le cadre du système expérimental.

### **Compte rendu du 4 décembre 2025 :**

Le 4 décembre, nous avons travaillé sur l'optimisation du système après le montage des différents éléments. Nous avons d'abord identifié et éliminé plusieurs fuites au niveau des réservoirs, en vérifiant les raccordements et l'étanchéité des connexions. Ces ajustements ont permis d'assurer un circuit fluide et sécurisé.

Une fois les problèmes d'étanchéité résolus, nous avons mis le processus en fonctionnement. Cette mise en marche a permis de vérifier le bon écoulement des fluides, la coordination des différents composants du système et la stabilité générale de l'installation en conditions réelles d'utilisation.



### **Compte rendu du 26 janvier 2026 :**

Ce jour-là, nous avons préparé une solution saline destinée aux mesures de conductivité à l'aide du conductimètre. Cette solution servira de référence pour l'analyse du fonctionnement du système.

Nous avons rencontré des difficultés au niveau de la programmation de la carte Arduino, en raison du nombre important de paramètres à intégrer, notamment le capteur de niveau et le

conductimètre. La gestion simultanée de ces capteurs a nécessité des ajustements dans le code afin d'assurer une communication et un contrôle corrects des différents éléments.

Après plusieurs essais et corrections, nous avons réussi à résoudre le problème de programmation. Nous avons finalement pu mettre la pompe en fonctionnement, ce qui a permis de valider le contrôle du système par l'Arduino.



### **Compte rendu du 28 janvier 2026 :**

---

Le 28 janvier, nous avons réalisé une première régulation du système, portant sur le contrôle du niveau. Le principe défini était le suivant : les deux pompes se mettent en marche lorsque la distance mesurée par le capteur de niveau est supérieure à **4 cm**, et elles s'arrêtent lorsque cette distance devient inférieure ou égale à **4 cm**.

Nous avons développé un programme Arduino permettant d'implémenter cette logique de régulation. En parallèle, nous avons effectué le branchement du capteur de niveau sur la carte Arduino : la borne **GND** du capteur a été reliée au **GND** de l'Arduino, la borne **VCC** au **5 V**, et les broches **Echo** et **Trig (signal)** ont été connectées aux entrées numériques **D2** et **D3** de l'Arduino.

Ces étapes ont permis de mettre en place une régulation automatique de niveau et de valider le bon fonctionnement du capteur associé au système.

### **LE CODE POUR LE CAPTEUR DE NIVEAU :**

```
// =====  
// CAPTEUR ULTRASON  
// =====
```

```

const int trigPin = 2;
const int echoPin = 3;

// =====
// POMPES (relais)
// =====
const int RELAY1_PIN = 7; // pompe 1
const int RELAY2_PIN = 8; // pompe 2

#define RELAY_ON HIGH
#define RELAY_OFF LOW

// =====
// SEUIL DE NIVEAU
// =====
const float SEUIL_CM = 3.0; // seuil unique à 3 cm

// -----
// Fonctions pompes
// -----
void pumpsOn() {
    digitalWrite(RELAY1_PIN, RELAY_ON);
    digitalWrite(RELAY2_PIN, RELAY_ON);
}

void pumpsOff() {
    digitalWrite(RELAY1_PIN, RELAY_OFF);
    digitalWrite(RELAY2_PIN, RELAY_OFF);
}

// -----
// Mesure distance (cm)
// -----
float readDistanceCm() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    unsigned long duree = pulseIn(echoPin, HIGH, 30000UL); // timeout 30 ms

    if (duree == 0) return -1.0; // erreur mesure

    return duree * 0.0343 / 2.0;
}

```

```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  pinMode(RELAY1_PIN, OUTPUT);
  pinMode(RELAY2_PIN, OUTPUT);

  pumpsOff();

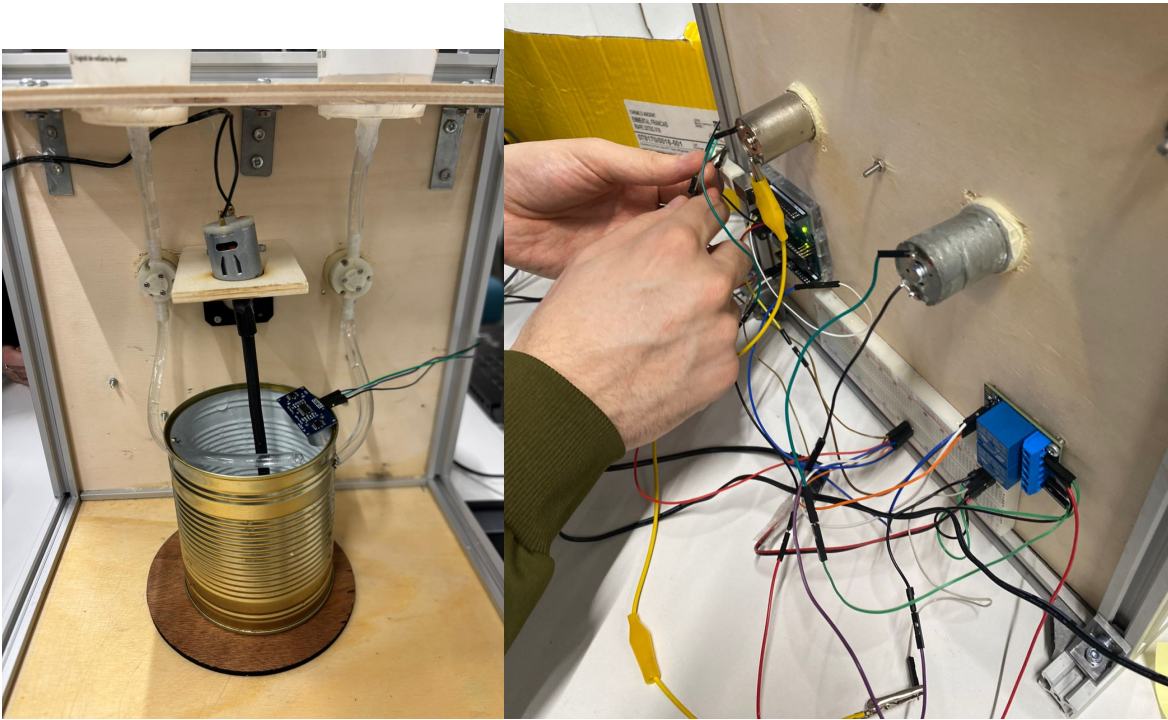
  Serial.begin(9600);
  Serial.println("Regulation niveau simple (seuil 3 cm)");
}
```

```
void loop() {
  float distance = readDistanceCm();

  Serial.print("Distance = ");
  Serial.print(distance);
  Serial.println(" cm");

  // ----- REGULATION SIMPLE -----
  if (distance >= SEUIL_CM) {
    // Niveau pas encore haut -> on remplit
    pumpsOn();
    Serial.println("Pompes ON");
  } else {
    // Niveau haut -> on stop
    pumpsOff();
    Serial.println("Pompes OFF");
  }

  delay(200); // petite pause pour stabilité
}
```



### **Compte rendu du 29 janvier 2026 :**

Le 29 janvier, nous avons mis en place une deuxième régulation du système, combinant le **contrôle du niveau** et celui de la **conductivité**.

Concernant le niveau, la consigne définie est la suivante : lorsque la distance mesurée par le capteur est **supérieure à 4 cm**, les deux pompes se mettent en marche. En revanche, lorsque la distance devient **inférieure ou égale à 4 cm**, les pompes s'arrêtent immédiatement afin d'éviter un débordement ou un fonctionnement non contrôlé.

Pour la régulation de la conductivité, nous avons utilisé deux fluides (**préparés et mesurés le 26 janvier dans le laboratoire de biologie/chimie du FabLab**) :

- une **solution saline** ayant une conductivité de **12,3 mS/cm**,
- de l'**eau distillée** avec une conductivité d'environ **1,5 µS/cm** (proche de zéro).

L'objectif est d'atteindre une conductivité de **7,2 mS/cm** dans le réacteur.

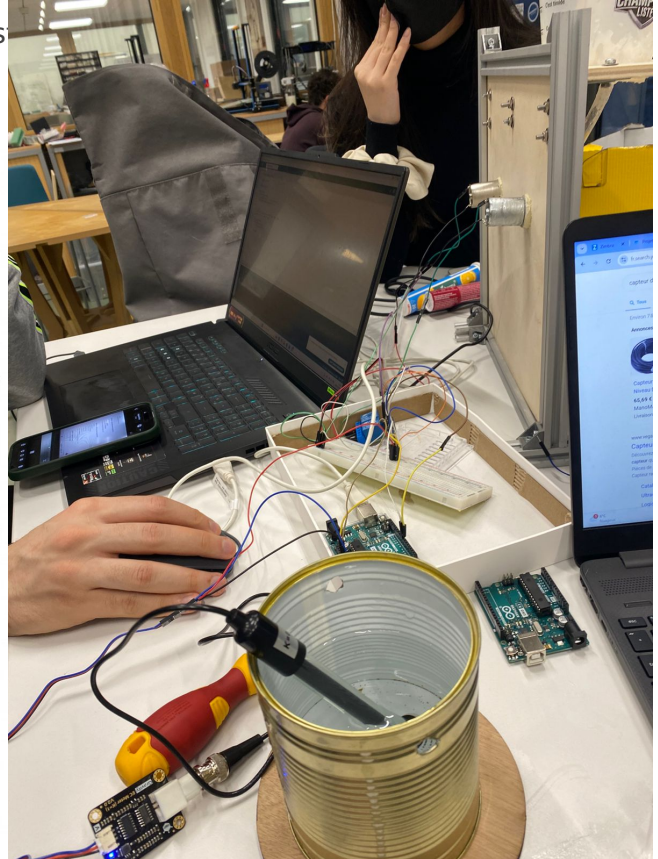
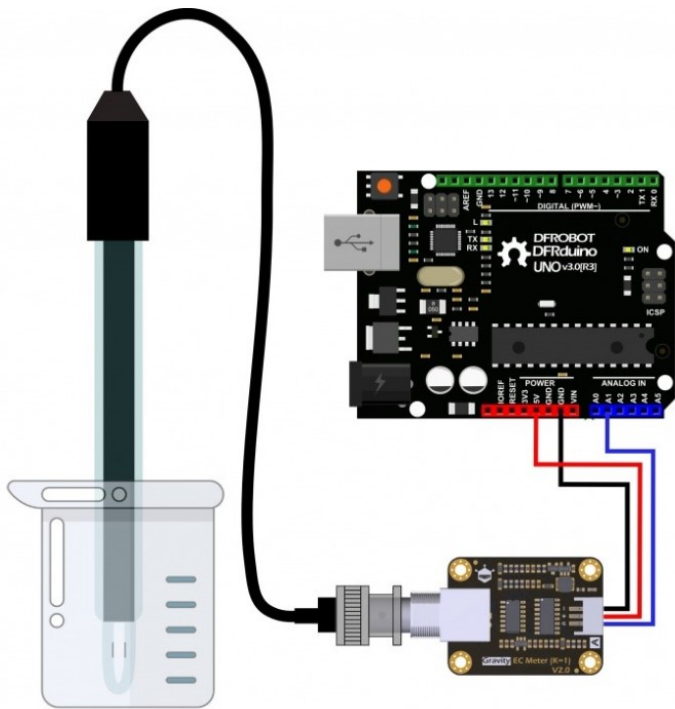
La logique de contrôle est la suivante :

- Si la conductivité mesurée est **supérieure à 7,2 mS/cm**, la pompe alimentant la solution saline est arrêtée et la pompe d'eau distillée fonctionne afin de diluer la solution.

- Si la conductivité est **inférieure à 7,2 mS/cm**, la pompe de solution saline est activée pour augmenter la conductivité.
- Lorsque la conductivité atteint la valeur cible de **7,2 mS/cm**, les deux pompes sont arrêtées.

Ainsi, les pompes peuvent s'arrêter dans deux cas :

1. lorsque le niveau atteint la limite fixée (distance  $\leq 4$  cm).
2. lorsque la conductivité cible de 7,2 mS/cm est atteinte.



Ce schéma montre le branchement d'un capteur de conductivité (EC meter) à une carte Arduino Uno pour mesurer la conductivité d'une solution liquide. La sonde de conductivité est immergée dans le bécher contenant la solution et reliée au module électronique du capteur, qui traite le signal. Le module est connecté à l'Arduino via le 5 V et le GND pour l'alimentation, tandis que le signal de mesure est transmis vers une entrée analogique de la carte. En parallèle, un capteur de niveau était également utilisé dans le système afin de surveiller la hauteur du liquide.

Pour faciliter les connexions entre les différents composants (capteurs, alimentation et Arduino), nous avons utilisé une **breadboard**, ce qui a permis d'organiser le câblage et d'effectuer les modifications plus facilement. L'Arduino peut ainsi lire la conductivité et le niveau du liquide pour assurer le contrôle et la régulation du procédé.

## LE CODE POUR LE CAPTEUR DE NIVEAU ET LE CONDUCTIMETRE :

```
#include "DFRobot_ECPRO.h"

// =====
// PINS
// =====
// Ultrason
const int trigPin = 2;
const int echoPin = 3;

// Pompes (relais)
const int PUMP_SALT_PIN = 7; // pompe solution saline
const int PUMP_FRESH_PIN = 8; // pompe eau distillée

// Conductivité
#define EC_PIN A0

// =====
// RELAIS (selon vos tests)
// =====
#define RELAY_ON HIGH
#define RELAY_OFF LOW

// =====
// OBJECTIFS
// =====
// Conductivité cible
const float EC_SETPOINT_mS = 7.2; // mS/cm
const float EC_BAND_mS = 0.2; // +/- 0.2 mS/cm (à ajuster)

// Sécurité niveau (distance en cm)
const float MAX_LEVEL_DISTANCE_CM = 4.0; // si distance <= 4 cm => STOP pompes

// =====
// TIMING (pour dosage)
// =====
const unsigned long PULSE_MS = 800; // temps ON pompe (dose)
const unsigned long SETTLE_MS = 800; // temps OFF pour mélange/lecture

// =====
// OBJET ECPRO
// =====
DFRobot_ECPRO ec;
```

```

// =====
// UTILITAIRES POMPES
// =====
void pumpsOff() {
    digitalWrite(PUMP_SALT_PIN, RELAY_OFF);
    digitalWrite(PUMP_FRESH_PIN, RELAY_OFF);
}

void saltOn() {
    digitalWrite(PUMP_SALT_PIN, RELAY_ON);
    digitalWrite(PUMP_FRESH_PIN, RELAY_OFF);
}

void freshOn() {
    digitalWrite(PUMP_SALT_PIN, RELAY_OFF);
    digitalWrite(PUMP_FRESH_PIN, RELAY_ON);
}

// =====
// MESURE DISTANCE (cm)
// =====
float readDistanceCm() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    unsigned long duree = pulseIn(echoPin, HIGH, 30000UL); // 30 ms timeout
    if (duree == 0) return -1.0; // invalide

    return (duree * 0.0343) / 2.0; // cm
}

// =====
// MESURE CONDUCTIVITE (mS/cm)
// =====
float readEC_mS_cm() {
    // Moyenne pour stabiliser un peu
    uint32_t sum = 0;
    const int N = 10;
    for (int i = 0; i < N; i++) {
        sum += analogRead(EC_PIN);
        delay(5);
    }
    uint16_t raw = sum / N;
}

```

```

// Conversion en mV (référence 5V)
uint16_t mV = (uint32_t)raw * 5000 / 1024;

// Librairie renvoie en µS/cm
float ec_us_cm = ec.getEC_us_cm(mV);

// Convertir en mS/cm
return ec_us_cm / 1000.0;
}

void setup() {
  // Ultrason
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Pompes
  pinMode(PUMP_SALT_PIN, OUTPUT);
  pinMode(PUMP_FRESH_PIN, OUTPUT);
  pumpsOff();

  // Conductivité
  Serial.begin(115200);
  ec.setCalibration(1); // mettez votre K calibré si vous l'avez
  Serial.print("Calibration K = ");
  Serial.println(ec.getCalibration());

  Serial.println("Demarrage regulation conductivite + securite niveau");
}

void loop() {
  // 1) Sécurité niveau (prioritaire)
  float dist = readDistanceCm();
  if (dist > 0) {
    Serial.print("Distance = ");
    Serial.print(dist, 1);
    Serial.print(" cm | ");
  } else {
    Serial.print("Distance = invalide | ");
  }

  if (dist > 0 && dist <= MAX_LEVEL_DISTANCE_CM) {
    pumpsOff();
    Serial.println("NIVEAU HAUT (<=4cm) -> POMPES OFF");
    delay(300);
    return; // on ne fait rien d'autre
  }
}

```

```

// 2) Mesure conductivité
float ec_mS = readEC_mS_cm();
Serial.print("EC = ");
Serial.print(ec_mS, 3);
Serial.println(" mS/cm");

// 3) Décision
float low = EC_SETPOINT_mS - EC_BAND_mS;
float high = EC_SETPOINT_mS + EC_BAND_mS;

if (ec_mS < low) {
  // Trop faible -> ajouter saline
  Serial.println("EC trop basse -> dose SALINE");
  saltOn();
  delay(PULSE_MS);
  pumpsOff();
  delay(SETTLE_MS);
}
else if (ec_mS > high) {
  // Trop élevée -> ajouter eau douce
  Serial.println("EC trop haute -> dose EAU DOUCE");
  freshOn();
  delay(PULSE_MS);
  pumpsOff();
  delay(SETTLE_MS);
}
else {
  // Dans la bande -> rien
  Serial.println("EC OK -> pompes OFF");
  pumpsOff();
  delay(500);
}
}

```

## **Conclusion**

Cette soutenance a marqué l'aboutissement de notre travail sur la conception et la mise en place d'un système de régulation combinant le contrôle du niveau et de la conductivité. Le projet nous a permis de passer d'une idée théorique à une réalisation concrète, en intégrant des éléments mécaniques, hydrauliques et électroniques autour d'une commande par Arduino.

Avec plus de temps, nous aurions pu améliorer la robustesse du montage (meilleure étanchéité, fixation plus durable des composants) ainsi que l'optimisation du code pour rendre la régulation plus stable et plus précise. Une meilleure calibration des capteurs aurait également permis d'augmenter la fiabilité des mesures.

Ce projet nous a surtout appris à travailler en équipe, à résoudre des problèmes techniques imprévus (fuites, erreurs de câblage, difficultés de programmation) et à adapter notre conception en cours de route. Nous avons retenu l'importance des tests progressifs, de la communication entre les membres du groupe et du lien entre théorie et pratique dans la réalisation d'un procédé expérimental.

---

Revision #26

Created 10 October 2025 12:28:48 by Ben Salha Firas

Updated 2 February 2026 12:09:47 by Kuta Sara