

Entreprise SMDW

# Rapport de Stage



**ECAM**  
Brussels Engineering School

DE PATOUL Benoît 5MEO ECAM  
03/10/2016

## CONTENU

---

1. Objectif et Introduction.....	2
2. Présentation de l'entreprise .....	3
2.1. Introduction et historique .....	3
2.2. Organigramme.....	3
2.3. Domaines.....	4
3. Projet .....	6
3.1. Présentation.....	6
3.2. Définitions .....	8
3.2.1. Pot catalytique .....	8
3.2.2. Catalyseur.....	11
3.2.3. Cérium ( <i>Ce</i> ).....	11
3.2.4. Arduino.....	12
3.3. Préparation.....	12
3.4. Expérimentations.....	13
3.4.1. Interactions et connexions des différents composants.....	13
3.4.2. Code Arduino et expériences .....	16
4. Conclusion.....	25
5. Annexes .....	26
5.1. Images des expériences .....	26
5.2. Calibrer le temps de fonctionnement de la pompe .....	27
5.3. Plan d'alimentation.....	28

# 1. OBJECTIF ET INTRODUCTION

---

L'objectif de ce rapport est de présenter le stage réalisé afin d'obtenir ma formation de Master en ingénieur électronique. Une présentation de l'entreprise et du projet réalisé seront abordés.

Le rapport a été écrit dans le cadre d'un stage d'immersion en entreprise d'une durée de 6 semaines comprises entre le 22/09/2016 et 03/11/2016. Le stage a été réalisé dans l'entreprise SMDW avec comme maître de stage M. Mertens et comme superviseur de stage M. Combéfis.

Le rapport sera divisé en 4 chapitres :

1. Objectif et introduction
2. Présentation de l'entreprise
3. Présentation du projet
4. Conclusion

Le deuxième chapitre donnera un aperçu global de l'entreprise et sera divisé en 3 sections :

- 2.1. Introduction et historique
- 2.2. Organigramme
- 2.3. Domaines

L'entreprise étant petite, le descriptif sera assez bref.

Le troisième chapitre abordera le projet réalisé dans l'entreprise et sera divisé en 4 sections :

- 3.1. Présentation
- 3.2. Définitions
- 3.3. Préparation
- 3.4. Expériences à la réalisation du projet

Le quatrième chapitre sera la conclusion du rapport. Il apportera les objectifs atteints, et les améliorations possibles du projet.

## 2. PRÉSENTATION DE L'ENTREPRISE

---

### 2.1. INTRODUCTION ET HISTORIQUE

---

La société SMDW SPRL a été créée en 1993 par M. Etienne Mertens. L'objectif de la société est d'accélérer la réinsertion des personnes handicapées à la vie quotidienne

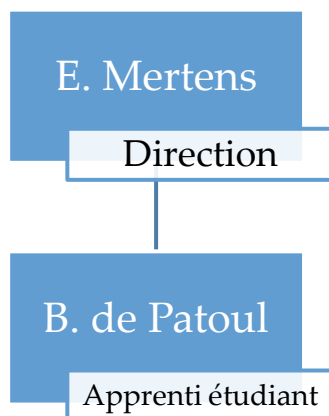
Le premier projet est de demander à André Malherbe (triple champion du monde de motocross devenu tétraplégique suite à un grave accident de moto au Paris-Dakar 88) de piloter une voiture entièrement adaptée à lui à l'occasion du Marathon Shell. Le véhicule est piloté avec la tête et avec la bouche, le tout assisté par de l'électronique.

Le projet est un succès : André Malherbe sort triomphant de l'épreuve avec ce prototype ayant une consommation de 1L pour 600km.

Source : <http://www.smdw.be/SMDW/fr/index.htm>

### 2.2. ORGANIGRAMME

---



## 2.3. DOMAINES

---

Les domaines de la société sont :

- A. l'adaptation des véhicules aux personnes handicapées ;
- B. les fauteuils roulants ;
- C. les voitures Burton.

- A. L'adaptation des véhicules roulants comprend l'adaptation du volant, des pédales, des sièges, l'application d'une rampe d'accès, un plancher décaissé, etc.



FIGURE 1

Source : <http://www.smdw.be/adaptations/Fr/index.html>

- B. Le domaine des fauteuils concerne la réparation, la construction sur mesure et l'application de « hemilead ». Hemilead est un levier de commande unilatéral à double effet. L'aller comme le retour du levier propulse le fauteuil roulant, rendant le déplacement bien plus optimal et efficace. La conduite du fauteuil est également simplifiée : avancer, reculer, freiner, tourner ... tout se fait d'une seule main. L'hémilead a été conçu pour permettre aux personnes souffrant d'hémiplégie de pouvoir se déplacer de manière indépendante. L'hémiplégie est une paralysie d'une ou plusieurs parties du corps d'un seul côté (droit ou gauche). Les causes sont nombreuses comme par exemple : une thrombose, un accident vasculaire cérébrale (AVC) ...

C. Le domaine des voitures Burton concerne la vente de pièces détachées et de voitures.



FIGURE 2

Source : <http://www.leblogauto.com/2009/11/essai-2cv-burton-quune-deuche.html>

## 3. PROJET

---

### 3.1. PRÉSENTATION

---

Lorsque je suis entré pour la première fois dans l'entreprise (un garage pour être plus précis), je suis tombé sur M. Mertens. Il m'a fait visiter son atelier où j'ai pu découvrir des voitures Burton pour la première fois, différents fauteuils roulants, et même un avion. Ensuite on est allé à son bureau où il m'a expliqué le projet qu'il avait en tête.

Actuellement, les fabricants de voitures utilisent le pot catalytique (définition page 8) afin respecter les taux légaux d'émission des gaz nocifs pour la santé (monoxyde de carbone, hydrocarbures imbrûlés et oxyde d'azote) définies par les normes européennes. Le pot catalytique sert à réduire les émissions des gaz toxiques au détriment d'une perte de charge (donc une augmentation de consommation). De plus, il ne fonctionne qu'à une certaine température (environ 400°C) atteinte après avoir roulé plusieurs kilomètres.

Pour M. Mertens, ce système n'est pas idéal, car non seulement on consomme plus en transformant ces gaz toxiques, mais en plus on ne récupère pas l'énergie de cette transformation exothermique. L'idée de M. Mertens est de réduire les gaz toxiques avant d'arriver au pot catalytique. Comment faire ? En augmentant l'efficacité de la combustion dans la chambre à combustion du moteur.

Le projet est appelé : « optimisation de la combustion interne d'un moteur bicylindre Citroën ». Il consiste à obtenir une combustion plus complète dans la chambre à combustion du moteur. L'objectif est de diminuer les gaz nocifs pour la santé émis (respecter les normes européennes), et de récupérer leur énergie lors de leur combustion. Pour y arriver, on amènera un catalyseur le cérium (voir page 11) dans de la vapeur d'eau jusqu'à la chambre à combustion du moteur (voir figure 3). Lors de l'explosion, le catalyseur d'améliorera la combustion et donc de diminuer les gaz nocifs pour la santé parmi les gaz d'échappement. Cela entraînera une augmentation du rendement moteur.

La réalisation de tout le projet, fera l'objet d'un travail de fin d'études. En tant que stagiaire pour six semaines, mon travail consiste à préparer le terrain. J'ai été amené à m'occuper du transport du catalyseur. J'ai dû réaliser un système de régulation du niveau d'eau en ébullition (qui se chargera de transporter le catalyseur) dans un convecteur (voir images en annexe) en utilisant une thermistance. Le système sera adapté à une voiture Burton.

Pour atteindre l'objectif, j'ai utilisé :

- un arduino (définition page 12) ;
- un relais ;
- une thermistance ;
- une diode ;
- une pompe ;
- un régulateur Step Down ;
- un fusible ;
- une batterie ;
- une résistance.

Le fonctionnement complet du système est expliqué à la page 13, mais voici déjà un aperçu :

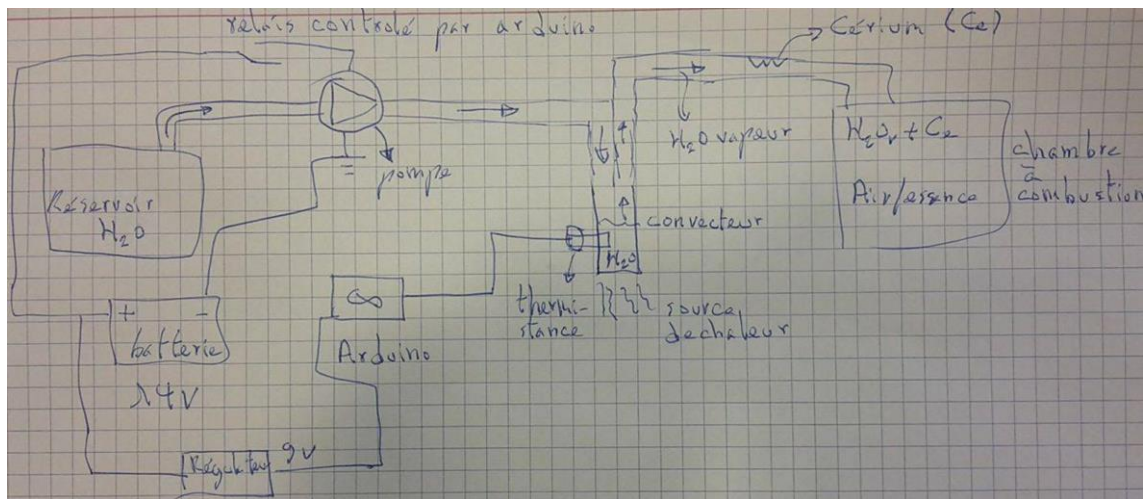


FIGURE 3

Ce rapport abordera diverses définitions afin de mieux comprendre le projet. Par la suite, la préparation et les différentes étapes du projet seront expliquées.



## 3.2. DÉFINITIONS

### 3.2.1. POT CATALYTIQUE

Voici une figure comprenant un moteur, un pot catalytique et un silencieux :

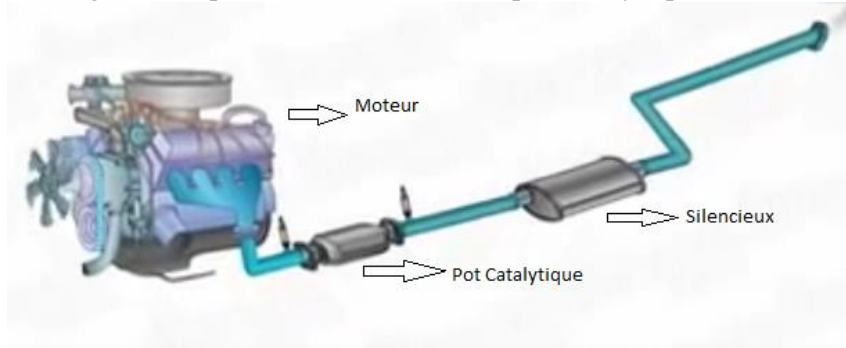


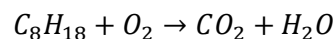
FIGURE 4

Source : [https://www.youtube.com/watch?v=W6dIsC\\_eGBI](https://www.youtube.com/watch?v=W6dIsC_eGBI)

Le carburateur du moteur assure le mélange air/carburant de la chambre à combustion. Lorsqu'on brûle le mélange, la combustion n'est pas parfaite et ne rejette donc pas seulement de l'eau et du dioxyde de carbone, mais aussi des gaz nocifs pour la santé. Les gaz d'échappements sont composés de :

- monoxyde de carbone ;
- dioxyde de carbone ;
- hydrocarbures imbrulés ;
- azote ;
- oxyde d'azote ;
- vapeur d'eau ;
- oxygène.

Le monoxyde de carbone ( $CO$ ) est un gaz incolore et inodore. La combinaison de carbone et d'oxygène est issue de la combustion incomplète des substances composées de carbone et constitue une substance toxique pour le système respiratoire. Sachant que l'essence est constituée majoritairement d'octane ( $C_8H_{18}$ ) :



Cette équation est correcte lorsque la combustion est parfaite, mais elle ne l'est pas. On ne retrouvera pas seulement du dioxyde de carbone à la fin de la combustion, mais aussi des gaz toxiques comme le monoxyde de carbone.

Lorsque ce gaz est inhalé et parvient dans le sang, il empêche les globules rouges de véhiculer correctement l'oxygène dans l'organisme et une mort par asphyxie peut survenir.

Les hydrocarbures imbrûlés ( $C_xH_y$ ) sont des composants constitués d'atomes d'hydrogène et de carbone. On en trouve un grand nombre résultant d'une combustion incomplète, souvent instables et sujets à des réactions chimiques entre eux : des hydrocarbures simples, des hydrocarbures aromatiques (ou benzéniques) monocycliques (benzène, toluène, xylène) ... Sous l'influence du rayonnement solaire, les hydrocarbures et l'oxyde d'azote se transforment en ozone. Dans les couches basses de l'atmosphère, l'ozone constitue une substance toxique dangereuse qui irrite les muqueuses et provoque des maux de tête et nausées.

L'oxyde d'azote est la forme gazeuse de l'azote. Sa formule est abrégée en  $NO_x$  car il existe plusieurs combinaisons possibles avec un nombre différent d'atomes :  $N_2O$ ,  $NO$ ,  $N_2O_3$ ,  $NO_2$ , etc ... Si l'oxyde d'azote est en contact avec l'eau, il produit des acides qui irritent les muqueuses et peuvent provoquer des lésions pulmonaires. L'oxyde d'azote est un gaz à effet de serre et nuit à la couche d'ozone protectrice dans les hautes couches de l'atmosphère.

L'élément après le moteur, est le pot catalytique (voir figure 6). Il a comme objectif de réduire un maximum ces gaz polluants en utilisant des catalyseurs. Celui d'une voiture à essence est dit à trois voies et celui d'une voiture à diesel est dit à "deux voies". Nous expliquerons celui à "trois voies" mais le principe reste le même pour les deux.

Le catalyseur est dit à « trois-voies » car il doit assurer trois réactions simultanées en utilisant des catalyseurs :

- une réduction des oxydes d'azote en azote et dioxyde de carbone ;
- une oxydation du monoxyde de carbone en dioxyde de carbone ;
- une oxydation des hydrocarbures imbrûlés en dioxyde de carbone et eau.

Les catalyseurs utilisés sont les métaux précieux platine, rhodium et palladium. Ceux-ci sont placés en forme de « nid d'abeille » afin d'obtenir le maximum de surface de contact avec les gaz. Cela permet de réduire ou d'oxyder un maximum de molécules de gaz. On retrouve aussi du dioxyde de cérium en fine couche dans le nid d'abeille qui permet de fournir de l'oxygène à la réaction si nécessaire ou d'en retenir en cas d'excès.

La figure suivante montre l'intérieur du pot catalytique :

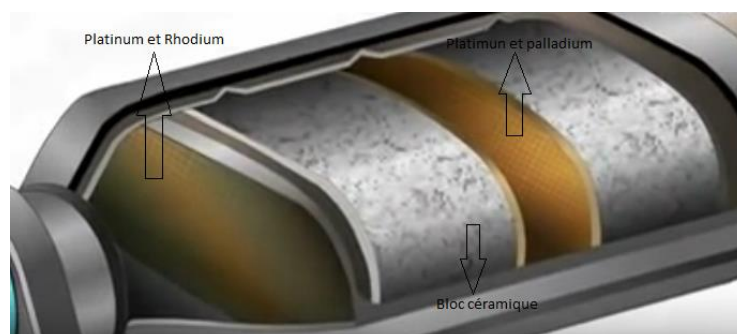
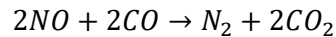


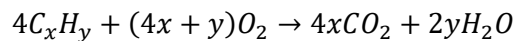
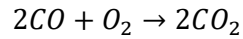
FIGURE 5

Source : [https://www.youtube.com/watch?v=W6dIsC\\_eGBI](https://www.youtube.com/watch?v=W6dIsC_eGBI)

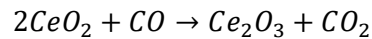
Lorsque les gaz pénètrent dans le premier bloc en céramique composé de platine et de rhodium, une réaction de réduction se produit sur les oxydes d'azote :



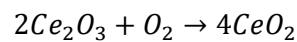
Lorsque les gaz traversent le deuxième bloc céramique composé de platine et de palladium, deux réactions d'oxydation se produisent :



Notons qu'en cas de manque d'oxygène, l'oxyde de cérium permet d'en fournir.



Ou en cas d'excès, d'en retenir :



On peut observer que les gaz sortant du pot catalytique sont moins toxiques après avoir traversés les catalyseurs.

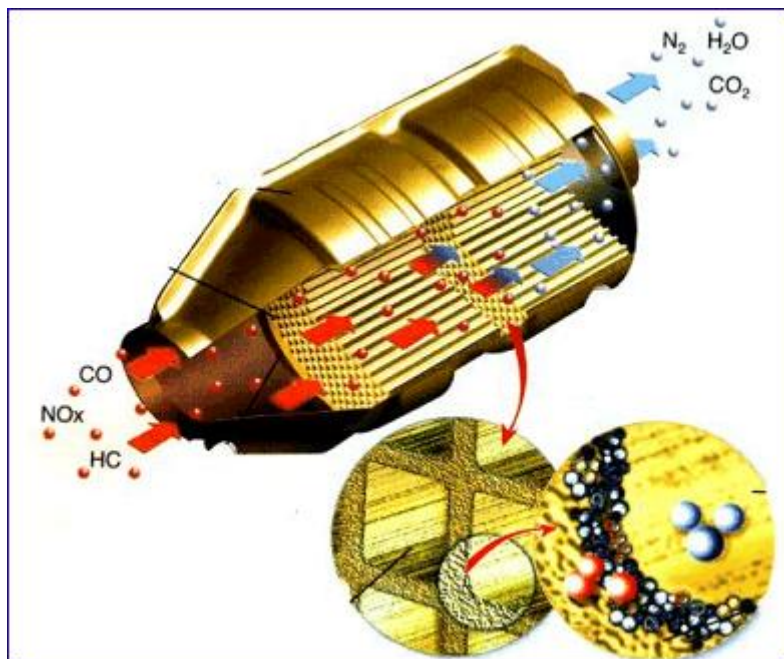


FIGURE 6

Source : <https://www.senat.fr/rap/r05-125/r05-12520.html>

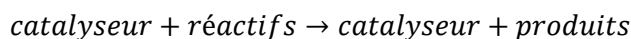
Après le pot catalytique on a un silencieux qui se charge d'atténuer le bruit du moteur. Pour cela, on utilise un système de rebondissements des ondes dans des compartiments qui sont en fonction de la puissance de celles-ci. On ne détaillera pas son fonctionnement car ce n'est pas nécessaire à la bonne compréhension du projet.

---

### 3.2.2. CATALYSEUR

---

Un catalyseur est une espèce chimique permettant d'augmenter la vitesse d'une réaction en diminuant l'énergie nécessaire pour la réaliser. Il peut être à l'état solide, liquide ou gazeux. Il ne figure pas dans l'équation chimique car il n'influence pas la composition du système dans l'état final. Le catalyseur présent au début de la réaction, se retrouvera à la fin de la réaction.

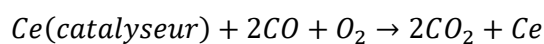


Il existe différents types de catalyse :

- catalyse homogène ; le catalyseur et les réactifs sont dans le même état, ils forment une seule phase ;
- catalyse hétérogène ; le catalyseur et les réactifs forment plusieurs phases.

Lors d'une réaction, le système peut évoluer différemment selon le catalyseur utilisé.

Dans le cas de notre projet, on compte utiliser du cérium comme catalyseur qui sera transporté par de la vapeur d'eau. La catalyse sera de type homogène car tous les composants de la réaction dans la chambre à combustion seront en phase gazeuse. L'utilité de ce catalyseur sera d'oxyder le monoxyde de carbone et de récupérer l'énergie dégagée de cette réaction.



Source : <http://lms.cours.fr/#/classe/ts/arbo/22898/opd/402440>

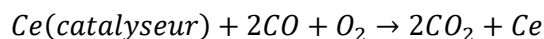
<http://www.web-sciences.com/fichests/fiche32/fiche32.php>

---

### 3.2.3. CÉRIUM (Ce)

---

Le cérium est la terre rare la plus abondante et c'est un métal gris argent. Dans le cadre du projet, on l'utilisera en tant que catalyseur d'oxydation :



Il n'existe pas de liste de catalyseurs en tant que soi, la seule manière de savoir si un élément est un catalyseur, c'est en réalisant des expériences dessus. Si après les expériences, on voit que le cérium ne fonctionne pas en tant que catalyseur, on compte se pencher sur un additif : l'oxyde de cérium IV ( $\text{CeO}_2$ ) ou aussi appelé « cérine » (utilisé dans pot catalytique). Que ce soit le cérium ou la cérine, l'élément sera transporté par de la vapeur d'eau vers la chambre à combustion.

---

### 3.2.4. ARDUINO

---

L'arduino est une carte électronique « open-source » (ouvert à tout le monde) qui permet de facilement utiliser le « software » (la partie codage) et le « hardware » (les composants physiques comme les PIN, des capteurs, etc). Il en existe plusieurs versions, mais celle qu'on utilise dans le cadre de ce projet, est l'arduino UNO. On peut assigner des tâches à la carte en écrivant le code approprié avec le programme arduino : <https://www.arduino.cc/en/Main/Software>

L'arduino est capable de lire une entrée comme par exemple : un doigt appuyant sur un bouton, un message internet, un capteur et la transformer en une sortie, c'est-à-dire respectivement : activer un moteur, répondre au message internet, allumer une LED.

Au démarrage du programme arduino, on se retrouve avec deux fonctions : « void setup » et « void loop ». La fonction « void setup » permet d'initialiser les variables que l'on va avoir besoin, elle est appelée qu'une seule fois au début du programme. Une fois passé la fonction « void setup », la fonction « void loop » prend le relais et tourne en boucle. C'est dans cette fonction qu'on écrit les tâches à réaliser continuellement par l'arduino. Dans notre cas, c'est contrôler régulièrement la tension sur la thermistance et fermer ou ouvrir le relais en fonction de la nécessité.

Un excellent site pour apprendre à débiter avec l'arduino gratuitement est :

<https://openclassrooms.com/courses/programmez-vos-premiers-montages-avec-arduino>

---

### 3.3. PRÉPARATION

---

Pour la préparation du projet j'ai acheté différents composants :

- un arduino UNO ;
- un régulateur de tension Step Down ;
- un boîtier;
- une pompe;
- un relais arduino;
- une breadboard.

J'ai dû aussi m'informer sur le fonctionnement du pot catalytique, le fonctionnement du système « optimisation de la combustion interne du moteur à explosion essence » expliqué dans le chapitre précédent et revoir la chimie de base.

## 3.4. EXPÉRIMENTATIONS

### 3.4.1. INTÉRACTIONS ET CONNEXIONS DES DIFFÉRENTS COMPOSANTS

Après l'achat des composants :

- un arduino ;
- un relais ;
- une thermistance ;
- une diode ;
- une pompe ;
- un régulateur Step Down ;
- un fusible ;
- une batterie ;
- une résistance.

J'ai connecté tous les composants ensemble et réalisé un schéma du circuit. La figure suivante reprend toutes les relations entre eux :

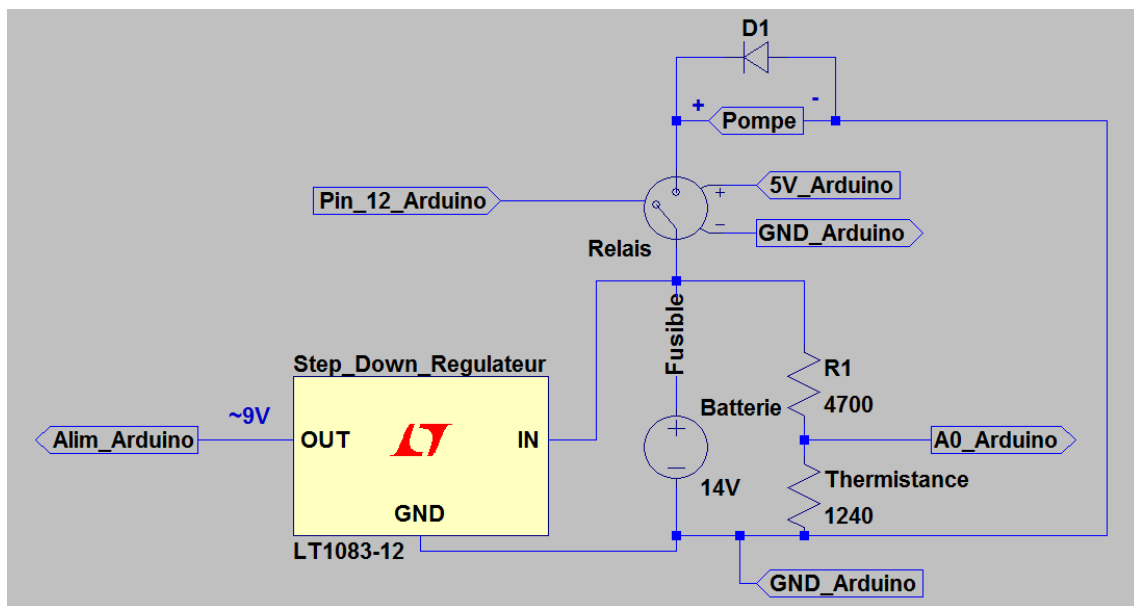


FIGURE 7

La source d'alimentation du circuit est la batterie de voiture délivrant 12V lorsqu'elle est à l'arrêt et 14V lorsque le moteur tourne (elle est en charge via l'alternateur). Elle permet d'alimenter l'ensemble du circuit : la pompe, l'arduino et la thermistance.

L'élément suivant après la batterie est le fusible. C'est un système de protection qui permet lors d'une surtension, de couper l'alimentation.

La tension optimale d'alimentation de l'arduino doit être située entre 7V et 12V et cela devient un problème lorsque le moteur tourne ( $14V > 12V$ ). La solution est de placer un régulateur de tension Step Down délivrant 9V en sortie.

Le relais est le composant qui permet d'ouvrir ou fermer le circuit d'alimentation qui activera la pompe. Il est alimenté en 5V par l'arduino et commandé par la PIN 12 de l'arduino.

La diode de protection qui est en parallèle, permet de protéger la pompe des surtensions lors de l'ouverture du relais.

La résistance R1 de  $4700\Omega$  est une résistance de protection qui permet de ne pas dépasser les 5V au point "A0\_Arduino". Ce point représente une connexion de sortie vers l'arduino.

La PIN A0 de l'arduino est un convertisseur analogique/numérique (CAN) qui permet de lire une tension analogique au point connecté du circuit, pour ensuite la convertir en numérique pour l'arduino. La tension maximale que peut lire cette PIN est de 5V. Au-delà, on risque d'endommager l'entrée, d'où la résistance R1 expliquée auparavant.

La thermistance est une résistance qui varie en fonction de la température. La variation de résistance entraîne selon la loi d'Ohm ( $U = R * I$ ) une variation de tension. Puisque la tension de la thermistance dépend de sa résistance qui elle-même dépend de la température, l'arduino peut calculer la température en détectant la tension sur la thermistance.

L'arduino ferme le relais (active la pompe) lorsque la température de la thermistance (chauffée par l'eau) est  $T \geq 100^{\circ}C$ . Pourquoi à  $100^{\circ}C$  ? Car lorsque la thermistance capte une température  $T \geq 100^{\circ}C$ , cela signifie qu'elle est hors de l'eau. L'arduino traduit cela comme un manque d'eau, et ferme le relais pour remplir à nouveau le convecteur. Pour chauffer l'eau et l'amener à ébullition, le convecteur utilisera comme source de chaleur le collecteur d'échappement.

Prenons un exemple en se basant sur le schéma du haut :

J'ai une thermistance qui à température ambiante, a une résistance de  $1240\Omega$  et à une température de  $100^{\circ}C$ , a une résistance de  $200\Omega$ . Elle est en série avec une résistance de  $4700\Omega$  et le tout est alimenté en 14V. Quel est la tension sur cette thermistance à température ambiante et à  $100^{\circ}C$  ?

Résolution :

A température ambiante, puisque les résistances sont en série :

$$R_{T.Ambiante} = 1240 + 4700$$

$$R_{T.Ambiante} = 5940\Omega$$

$$U = R * I \rightarrow 14 = 5940 * I$$

$$I_{T.Ambiante} = 0.0024A$$

Puisque les résistances sont en séries, le courant est le même pour les deux mais la tension est différente. La tension sur la thermistance à température ambiante est :

$$U_{T.Ambiante} = 1240 * 0.0024$$

$U_{T.Ambiante} = 2.976V$
---------------------------

A 100°C :

$$R_{100^{\circ}C} = 200 + 4700$$

$$R_{100^{\circ}C} = 4900\Omega$$

$$I_{100^{\circ}C} = \frac{14}{4900}$$

$$I_{100^{\circ}C} = 0.0028A$$

La tension sur la thermistance à 100°C est :

$$U_{100^{\circ}C} = 200 * 0.0028$$

$U_{100^{\circ}C} = 0.56V$
----------------------------

On conclut que la tension entre les deux températures est différente et quand on chauffe la thermistance, sa tension diminue.

Lorsque l'arduino lis une tension de  $\leq 0.56V$  sur la thermistance, il fermera le relais et la pompe sera alimentée.



---

### 3.4.2. CODE ARDUINO ET EXPÉRIENCES

---

Le code arduino sera l'intelligence du projet. Plusieurs versions ont été écrites afin de régler au fur et à mesure les différents problèmes rencontrés lors des expériences.

Le premier code affiché ci-dessous contient les deux fonctions classiques « void setup » et « void loop ».

Première version du code :

```
#define controle 12

void setup() {
  digitalWrite(controle,HIGH);
  Serial.begin(9600);
  pinMode(relais, OUTPUT);
}

void loop() {
  int lecture = analogRead(A0);
  float tension = lecture * (5.0/1023.0);

  if (tension>0 && tension <=2){
    Serial.println("LOW");
    Serial.println(tension);
    digitalWrite(controle,LOW);
  }
  if (tension>2){
    Serial.println("HIGH");
    Serial.println(tension);
    digitalWrite(controle,HIGH);
  }
}
```

La première fonction contient l'initialisation de la PIN 12 de l'arduino. Cette PIN est chargée d'envoyer le signal pour fermer ou ouvrir le relais.

La deuxième fonction contient comme premiers éléments : la lecture de la tension sur la thermistance via la PIN A0 et le calcul de la tension. La précision de la lecture est de 10 bits (donc 1024 valeurs :  $2^{10}$ ). Sachant que la PIN peut lire maximum 5V avec une précision de 10 bits, on obtient une précision en tension de  $5/1023 = 0.004883V$ . Je divise par 1023 car en informatique on compte le 0 parmi les 1024 valeurs. Par la suite, j'ai écrit deux conditions « if ». La première permet de fermer le relais lorsque la tension est entre 0V et 2V afin d'alimenter la pompe. La deuxième condition permet d'ouvrir le relais lorsque la tension est plus grande que 2V.

Lors des essais du circuit avec ce code, j'ai remarqué un **problème d'oscillation** entre la fermeture et l'ouverture du relais. Ceci était dû à une chute de tension comprise entre 10mV et 60mV lors de l'ouverture du relais et à un temps de stabilisation.

Puisque les conditions pour fermer ou ouvrir le relais s'activent autour de la valeur 2, cela produisait une oscillation.

Les figures suivantes montrent la chute de tension et le temps de stabilisation. Les valeurs sont des tensions en volts sur la thermistance et LOW=relais fermé, HIGH=relais ouvert :

LOW	HIGH	LOW
2.02	1.97	2.03
LOW	HIGH	LOW
2.03	1.97	2.03
HIGH	LOW	HIGH
2.03	1.97	2.03
HIGH	LOW	HIGH
2.00	1.97	2.01
HIGH	LOW	HIGH
2.11	1.88	2.11
HIGH	LOW	HIGH
2.11	1.88	2.11
HIGH	LOW	HIGH
2.12	1.88	2.11
HIGH	LOW	HIGH
2.12	1.88	2.11
HIGH	LOW	HIGH
2.11	1.88	2.11
HIGH	LOW	HIGH
2.11	1.88	2.11
HIGH	LOW	HIGH
2.11	1.88	2.11
HIGH	LOW	HIGH
2.11	1.88	2.11
HIGH	LOW	HIGH
2.11	1.88	2.11

Pour solutionner le problème d'oscillation, au lieu d'appliquer les conditions en fonction d'une valeur, je vais les appliquer en fonction d'une plage de valeurs qui se chevauchent. La version suivante du code contient la solution.

## Deuxième version du code :

```
#define relais 12

void setup() {
  pinMode(relais, OUTPUT);
  digitalWrite(relais,HIGH);
  Serial.begin(9600);
}

void loop() {

  int lecture = analogRead(A0);
  float tension = lecture * (5.0/1023.0);
  if (tension>0 && tension <2){
    while (tension < 2.03){
      lecture = analogRead(A0);
      tension = lecture * (5.0/1023.0);
      Serial.println("LOW");
      Serial.println(tension);
      digitalWrite(relais,LOW);
    }
  }
  if (tension>1.97){
    while (tension > 1.97){
      lecture = analogRead(A0);
      tension = lecture * (5.0/1023.0);
      Serial.println("HIGH");
      Serial.println(tension);
      digitalWrite(relais,HIGH);
    }
  }
}
```

Cette deuxième version du code contient les deux conditions « if » dans lesquelles j'ai rajouter des boucles « while » (tant que). Une boucle « while » permet d'obliger à l'arduino de rester dans la boucle tant que la condition pour y sortir n'a pas été remplie. Pour mieux comprendre ce que je veux dire, suivons le chemin par étapes qu'empreinte l'arduino dans le code :

1. L'arduino active la PIN 12, lui donne le nom de « relais » et la désigne en tant que sortie (envoyer des instructions).

```
#define relais 12

void setup() {
  pinMode(relais, OUTPUT);
  digitalWrite(relais,HIGH);
  Serial.begin(9600);
}
```

2. L'arduino lis la valeur de la tension au point connecté de la PIN A0 (dans notre cas, le thermistance) et la normalise. La variable résultante est appelée « tension »

```
int lecture = analogRead(A0);  
float tension = lecture * (5.0/1023.0);
```

3. Première condition « if » : si la « tension » est plus grande que 0 et est plus petite que 2, alors l'arduino rentre dans la condition.

```
if (tension>0 && tension <2){
```

4. En prenant l'hypothèse que la « tension » est plus grande que 0 et plus petite que 2, l'arduino rentre dans la condition et se retrouve dans la boucle « while » délimitée par des accolades. Celle-ci oblige l'arduino à y rester tant que la condition (*tension* < 2.03) est respectée. Une fois dans la boucle, l'arduino répète la séquence d'instructions suivantes : une lecture, une mise à jour de la variable « tension » et garder le relais fermé.

N'oublions pas que l'arduino se trouve dans cette boucle car la thermistance a capté une trop grande chaleur (elle se trouve hors de l'eau). Pour la refroidir on ferme le relais, ce qui allume la pompe et rempli d'eau le convecteur. La tension sur la thermistance augmente jusqu'à être plus grande que 2.03V et pouvoir sortir de la boucle.

```
while (tension < 2.03){  
  lecture = analogRead(A0);  
  tension = lecture * (5.0/1023.0);  
  Serial.println("LOW");  
  Serial.println(tension);  
  digitalWrite(relais,LOW);  
}
```

5. En prenant l'hypothèse que la « tension » est plus grande que 2.03, l'arduino quitte la boucle et vérifie la deuxième condition « if ». Puisque la « tension » est plus grande que 1.97, alors elle rentre dans la deuxième condition.

```
if (tension>1.97){
```

6. Une fois rentré dans la condition, l'arduino se retrouve dans la deuxième boucle « while ». Le fonctionnement est le même quand dans l'étape 4 avec comme seule différence, la condition pour rester dans la boucle (*tension* > 1.97).

```
while (tension > 1.97){  
  lecture = analogRead(A0);  
  tension = lecture * (5.0/1023.0);  
  Serial.println("HIGH");  
  Serial.println(tension);  
  digitalWrite(relais,HIGH);  
}
```

La solution au problème d'oscillation de la première version du code se trouve dans les conditions et à l'utilisation de boucles « while ». Dans la deuxième version du code je force l'arduino de garder le relais fermé et laisser la « tension » grimper jusqu'à atteindre 2.03. Lorsque la tension est atteinte, l'arduino quitte la première boucle, rentre dans la deuxième et ouvre le relais. A l'ouverture du relais, la « tension » de 2.03 chutera de par exemple 0.04. Mais puisque la condition pour rester dans la deuxième boucle est respectée (*tension* > 1.97) alors le relais **restera** fermé. Le relais n'oscillera plus et le code ne sera plus affecté par le temps de stabilisation de la tension.

Lors des essais avec la deuxième version du code, j'ai rencontré un **problème de température**. Lorsque je chauffais l'eau dans le convecteur, elle s'évaporait comme prévu. Mais quand la thermistance n'était plus immergée, au lieu de monter en température, elle diminuait. Si la température diminue, la tension sur la thermistance augmente ce qui, fausse mon code. Ce problème m'a obligé à restructurer tout le code et réaliser une troisième version du code.

### Troisième version du code :

```
#define relais 12
/*
 * On n'a pas la courbe de tension/température de la thermistance. On sait seulement
 * que la tension à 100°C de la thermistance est de 0.17V. Donc pour régler la température
 * à laquelle activer le relais, il faut diminuer ou augmenter la variable tensionThermistance.
 * Il ne faut pas perdre de vue que <0.17 permet d'augmenter la température à laquelle activer
 * le relais; et >0.17 permet de diminuer la température à laquelle on veut activer le relais.
 */
float tensionThermistance = 0.17; //Il suffit de jouer sur cette valeur pour calibrer l'activation du relais

void setup() {
  pinMode(relais, OUTPUT);
  digitalWrite(relais,HIGH);
  Serial.begin(9600);
}

void loop() {
  int lecture = analogRead(A0);
  float tension = lecture * (5.0/1023.0);
  digitalWrite(relais,HIGH);
  Serial.println(tension);
  if(tension <= tensionThermistance){
    while(tension <= tensionThermistance+0.01){
      lecture = analogRead(A0);
      tension = lecture * (5.0/1023.0);
      Serial.print("HIGH=");
      Serial.println(tension);
    }
    while(tension <= 0.23){
      lecture = analogRead(A0);
      tension = lecture * (5.0/1023.0);
      Serial.print("LOW=");
      Serial.println(tension);
      digitalWrite(relais,LOW);
    }
  }
}
```

Dans cette nouvelle version du code j'ai ajouté une nouvelle variable « tensionThermistance » qui permet de **calibrer la température d'activation du relais**. Lorsqu'on chauffe l'eau et on atteint les 100°C (« tensionThermistance » = 0.17) l'arduino rentre dans la première boucle « while » et y reste tant qu'il respecte la condition (elle est immergée dans l'eau bouillante). Lorsque la sonde n'est plus immergée la température qu'elle capte diminue jusqu'à atteindre 0.19 dans la variable « tension » (n'oublions pas que lorsque la température diminue, inversement la tension augmente sur la thermistance). L'arduino ne respectera plus la condition de la première boucle et rentrera donc dans la deuxième. Dans cette deuxième boucle, l'arduino remplit le convecteur d'eau ce qui, diminue la température et donc augmente la valeur de « tension » jusqu'à atteindre 0.23. L'arduino sort de la boucle et recommence à parcourir le code depuis le début de la fonction void loop(). Afin de **calibrer le temps de fonctionnement de la pompe**, il faut changer dans la condition du deuxième « while » la valeur 0.23. Plus on augmente la valeur, plus on augmente le temps de fonctionnement de la pompe.

Lors des essais avec cette nouvelle version du code, j'ai remarqué qu'il y avait trop d'intervenants externes qui influençaient la tension sur la thermistance lorsqu'elle était hors de l'eau : la puissance de la source de chaleur, le cuivre et certainement d'autres facteurs dont je n'ai pas remarqué. Il était donc très difficile de calibrer le code.

Afin d'éviter tous ces effets indésirables, j'ai décidé de changer de méthode : supprimer la thermistance et utiliser l'eau comme conducteur. Dans ce nouveau système nous n'utiliserons plus la température comme variable, mais l'eau en tant que conducteur.

L'idée est simple : lorsqu'il n'y aura plus d'eau pour conduire le courant, l'arduino le détectera et activera le relais. La thermistance sera remplacée par deux conducteurs isolés l'un de l'autre (voir figure 8 et 9). Lorsqu'ils sont immergés, l'eau sert de connexion entre les deux conducteurs et le courant peut circuler. Lorsqu'ils ne sont pas immergés, il n'y a pas de courant qui les traversent. Si le courant circule, c'est qu'il y a une différence de potentiel et l'arduino peut la détecter.

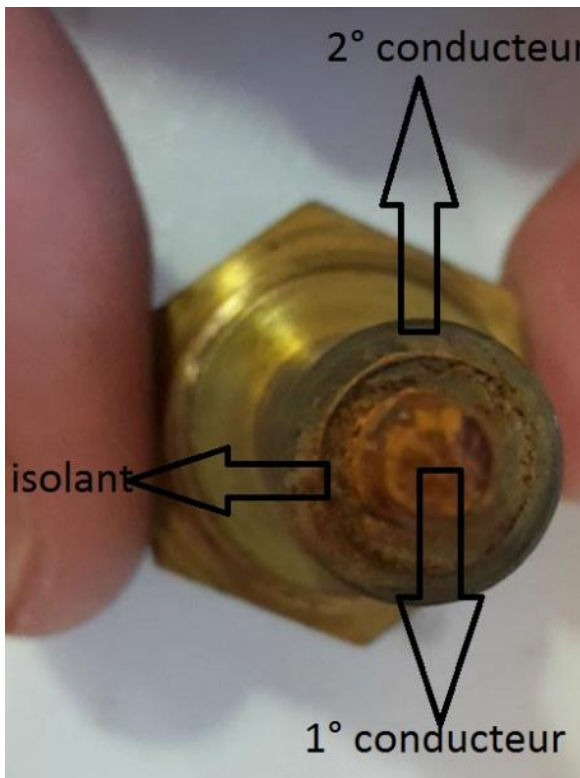


FIGURE 8



FIGURE 9

La quatrième version du code permet d'appliquer ce système.

#### Quatrième version du code :

```
#define relais 12
#define LED 10
int tempsPompe = 1000; //temps de fonctionnement de la pompe en ms

void setup() {
  pinMode(relais, OUTPUT);
  pinMode(LED, OUTPUT);
  digitalWrite(LED, LOW);
  digitalWrite(relais, HIGH);
  Serial.begin(9600);
}

void loop() {
  int lecture = analogRead(A0);
  float tension = lecture * (5.0/1023.0);
  delay(500);
  Serial.println(tension);
  if(tension >= 4){
    //Allumer pompe
    digitalWrite(relais, LOW);
    delay(tempsPompe);
    lecture = analogRead(A0);
    tension = lecture * (5.0/1023.0);
    Serial.println(tension);
    if(tension >= 4){
      while(true){
        //Allumer LED pour signaler un problème
        lecture = analogRead(A0);
        tension = lecture * (5.0/1023.0);
        Serial.println(tension);
        digitalWrite(relais, HIGH);
        digitalWrite(LED, HIGH);
        delay(2000);
        digitalWrite(LED, LOW);
        delay(2000);
      }
    }
  }
  else{
    //Couper la pompe
    digitalWrite(relais, HIGH);
  }
}
```

Après différents essais, ce nouveau système fonctionne correctement et est plus efficace que l'usage de la thermistance. Cette méthode va permettre d'apporter de nombreuses améliorations expliquées lors de la conclusion du rapport page 24. On peut changer le temps de fonctionnement de la pompe au début du code.



Avec cette nouvelle méthode, j'ai établi un nouveau plan d'alimentation :

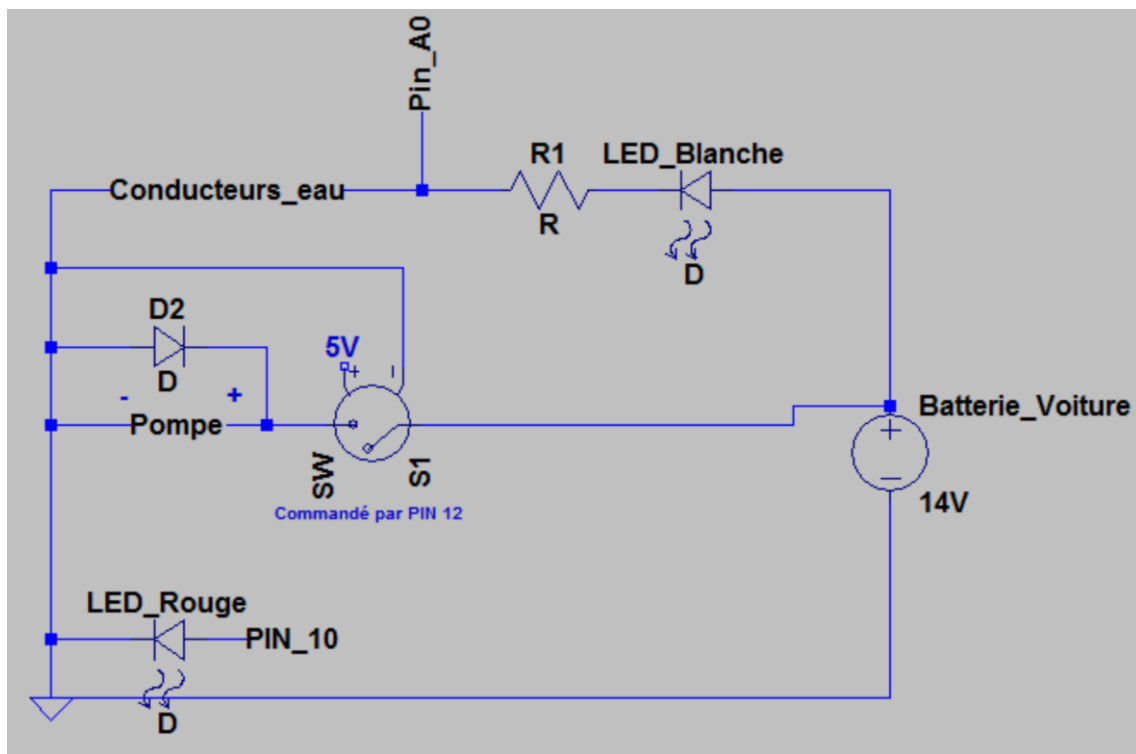


FIGURE 10

## 4. CONCLUSION

---

L'objectif du stage était de réussir à réaliser un boîtier permettant de réguler le niveau d'eau dans un convecteur. J'ai atteint cet objectif avec en plus des améliorations.

A la base, le projet devait être réalisé avec une thermistance. Mais après de nombreux essais et améliorations, le système n'était pas assez précis et fiable. D'autres problèmes se rajoutaient à chaque correction du système ce qui, m'a obligé à chercher une nouvelle méthode : l'utilisation de l'eau.

Après avoir réalisé les premiers essais, la nouvelle méthode utilisant l'eau comme conducteur semble prometteuse. Cette méthode permettra non seulement de fiabiliser le système, mais aussi de réduire grandement les coûts du système. La première version de la nouvelle méthode avait l'inconvénient d'avoir les conducteurs séparés par un isolant (voir figures 8 et 9). L'eau pouvait par fines couches relier les deux conducteurs même lorsqu'ils n'étaient plus immergés (l'eau n'étant pas totalement évaporée).

### Améliorations :

- Changer d'emplacement les conducteurs et mieux les isoler. Pour cela il faudrait deux conducteurs situés l'un en face de l'autre dans le convecteur (voir figure 11) séparés par l'air lorsqu'ils ne sont pas immergés. Cela permettra d'avoir une fiabilité du système à 100% !

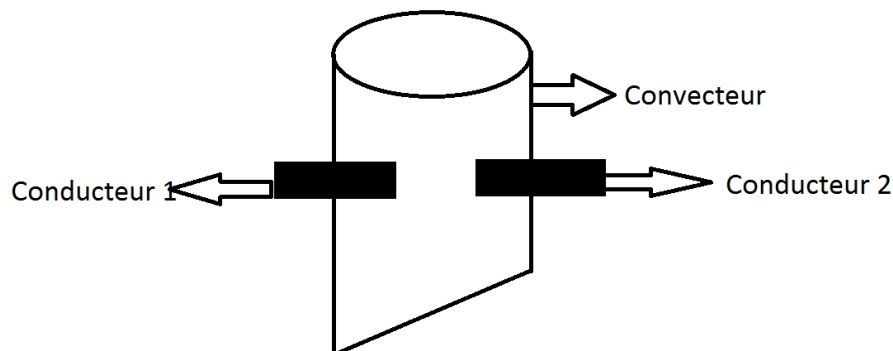


FIGURE 11

- Utilisant l'eau comme conducteur, on peut réduire les coûts du système en remplaçant l'arduino par un relais temporisé.

Ça n'a pas été un stage facile, mais j'ai appris beaucoup de choses et sur tout ce que je dois m'améliorer.

Je remercie mon maître de stage M. Mertens pour sa patience et son aide à la réalisation du rapport et de la présentation. Je remercie mon superviseur de stage M. Combéfis pour son appui tout au long du stage.

## 5. ANNEXES

### 5.1. IMAGES DES EXPÉRIENCES

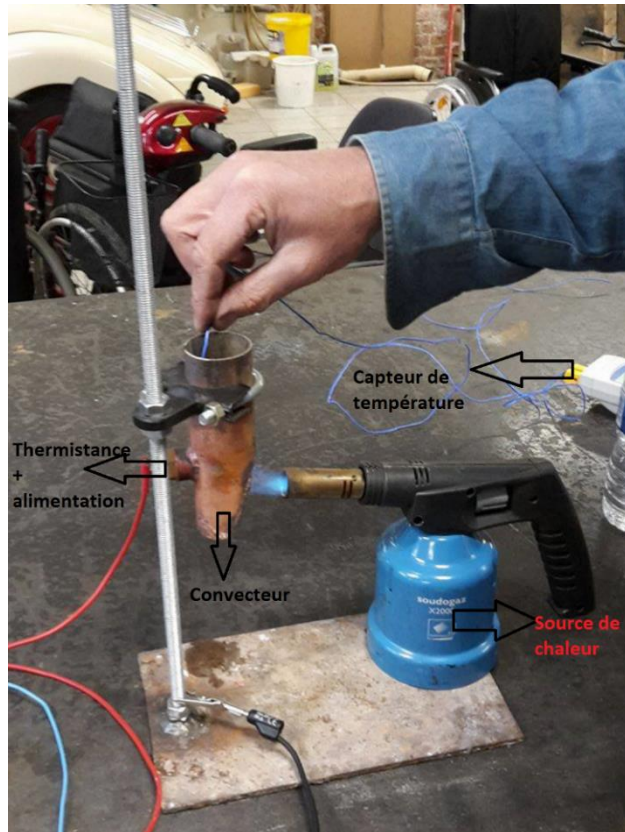


FIGURE 12

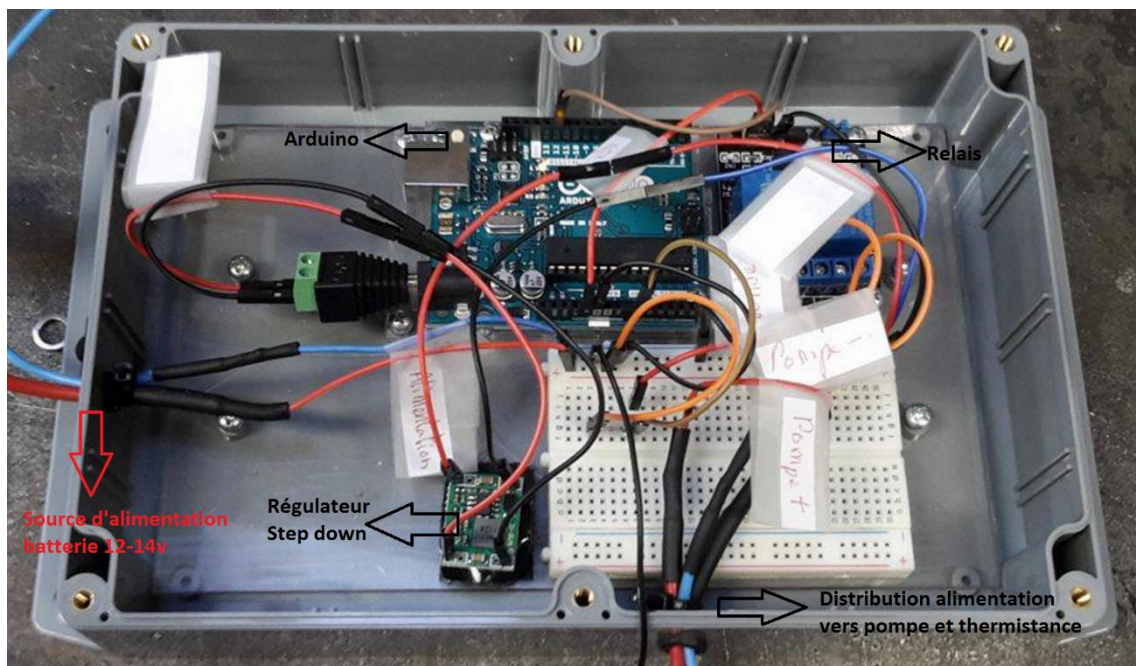


FIGURE 13

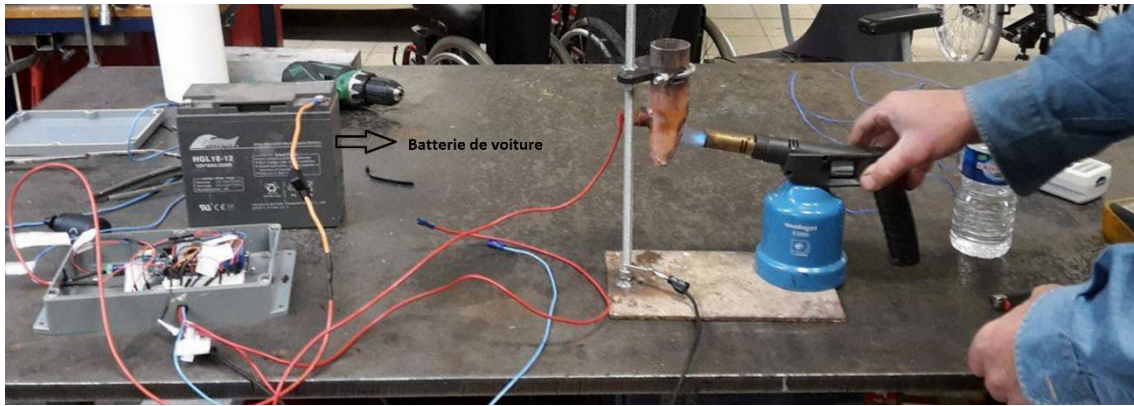


FIGURE 14

## 5.2. CALIBRER LE TEMPS DE FONCTIONNEMENT DE LA POMPE

---

Etapes :

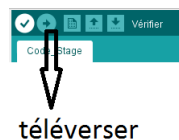
1. Installer programme arduino qui peut être télécharger dans :

<https://www.arduino.cc/en/Main/Software>

2. Ouvrir le fichier « Code Stage » avec le programme « arduino ».
3. Régler le temps de fonctionnement de la pompe en changeant la valeur du temps qui est en millisecondes :

```
int tempsPompe = 1000; //temps de fonctionnement de la pompe en ms
```

4. Une fois le temps régler, connecter l'arduino à l'ordinateur via un câble USB
5. Cliquez sur téléverser sur le coin en haut à gauche :



Si cela ne fonctionne pas (ce qui peut arriver), déconnecter et reconnecter l'USB  
Et ensuite réessayer de téléverser.

### 5.3. PLAN D'ALIMENTATION

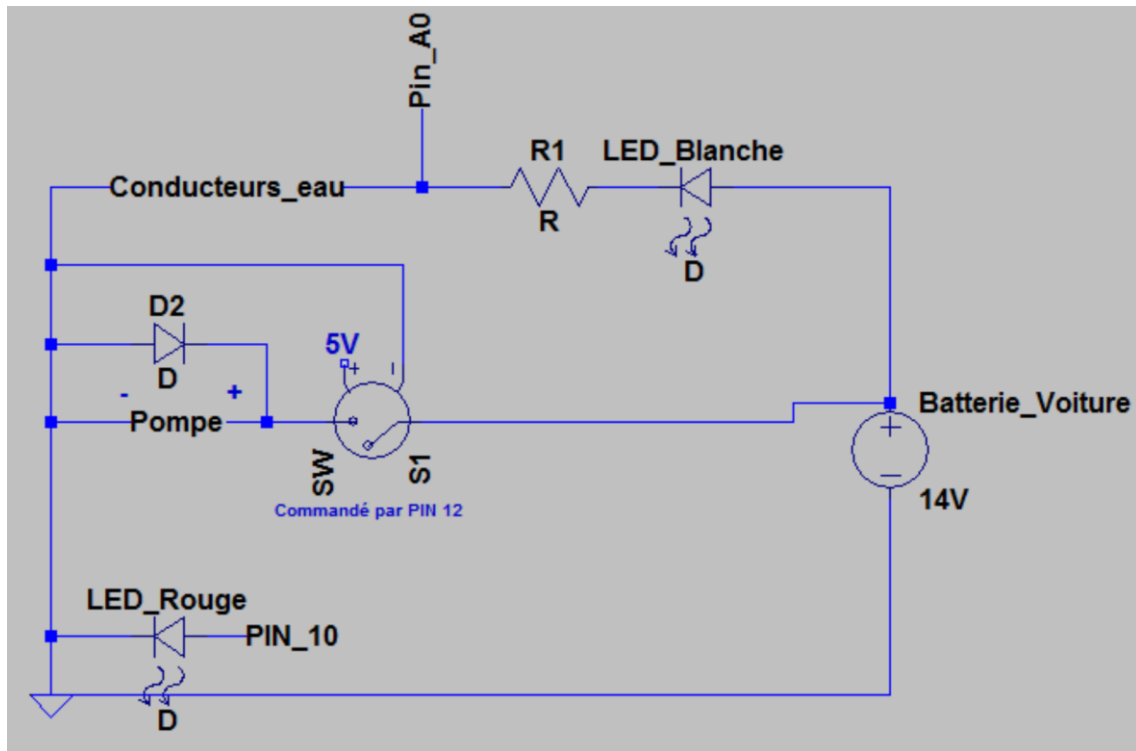


FIGURE 15