

# Projet PMF



**RAPPORT FINAL**  
GROUPE A2 ORAN

## Contenu

Introduction : .....	2
Plan prévisionnel : .....	3
Plan réel : .....	4
Plan du dispositif : .....	5
Formules : .....	5
Comment ça marche ? .....	6
Architecture utilisée (Code JAVA) : .....	9
Bilans personnels : .....	10

## Introduction :

Le principal objectif de ce projet est de transformer un simple mini-frigo USB en un système qui est automatisé. À partir d'une valeur de consigne, il doit être capable de réguler seul sa température afin d'atteindre cette même consigne, tout en détectant le taux d'humidité pour ainsi prévenir les risques de condensation.

## Plan prévisionnel :

Mehdi Belhadj					
Taches à accomplir	Dim 19	Mard 20	Mer 21	Jeudi 22	Dim 25
Montage du circuit	X	X	X		
Code Arduino					
Interface Eclipse					
Communication Ard/Eclipse			X		
Affichage graphs et variables					
MOSFET et comm courant					
Reporting			X	X	
Poster					
Merouane Bouderbala					
Taches à accomplir	Dim 19	Mard 20	Mer 21	Jeudi 22	Dim 25
Montage du circuit					
Code Arduino					
Interface Eclipse	X	X			
Communication Ard/Eclipse			X	X	
Affichage graphs et variables			X		
MOSFET et comm courant			X	X	
Reporting				X	
Poster					
Morad Djellouli					
Taches à accomplir	Dim 19	Mard 20	Mer 21	Jeudi 22	Dim 25
Montage du circuit	X	X	X		
Code Arduino		X	X		
Interface Eclipse					
Communication Ard/Eclipse					
Affichage graphs et variables					
MOSFET et comm courant			X	X	
Reporting					
Poster				X	X
Rayan Lakhdar					
Taches à accomplir	Dim 19	Mard 20	Mer 21	Jeudi 22	Dim 25
Montage du circuit					
Code Arduino					
Interface Eclipse		X	X		
Communication Ard/Eclipse			X	X	
Affichage graphs et variables			X		
MOSFET et comm courant			X	X	
Reporting					
Poster				X	X

## Plan réel :

Mehdi Belhadj					
Taches à accomplir	Dlm 19	Mard 20	Mer 21	Jeudi 22	Dlm 25
Montage du circuit	X	X	X		
Code Arduino					
Interface Eclipse					
Communication Ard/Eclipse			X		
Affichage graphs et variables					
MOSFET et comm courant					
Reporting			X	X	
Poster					
Merouane Bouderbala					
Taches à accomplir	Dlm 19	Mard 20	Mer 21	Jeudi 22	Dlm 25
Montage du circuit					
Code Arduino					
Interface Eclipse	X	X	X		
Communication Ard/Eclipse			X	X	
Affichage graphs et variables			X		X
MOSFET et comm courant		X	X	X	
Reporting			X	X	
Poster					
Morad Djellouli					
Taches à accomplir	Dlm 19	Mard 20	Mer 21	Jeudi 22	Dlm 25
Montage du circuit	X	X	X		
Code Arduino		X	X		
Interface Eclipse					
Communication Ard/Eclipse					
Affichage graphs et variables					
MOSFET et comm courant		X	X	X	
Reporting					
Poster			X	X	X
Rayan Lakhdar					
Taches à accomplir	Dlm 19	Mard 20	Mer 21	Jeudi 22	Dlm 25
Montage du circuit					
Code Arduino					
Interface Eclipse		X	X		
Communication Ard/Eclipse			X	X	
Affichage graphs et variables			X		X
MOSFET et comm courant		X	X	X	
Reporting					
Poster			X	X	X

Prévisionnel	X
Réel	X

## Plan du dispositif :

La modélisation a été effectuée à l'aide du logiciel Fritzing :

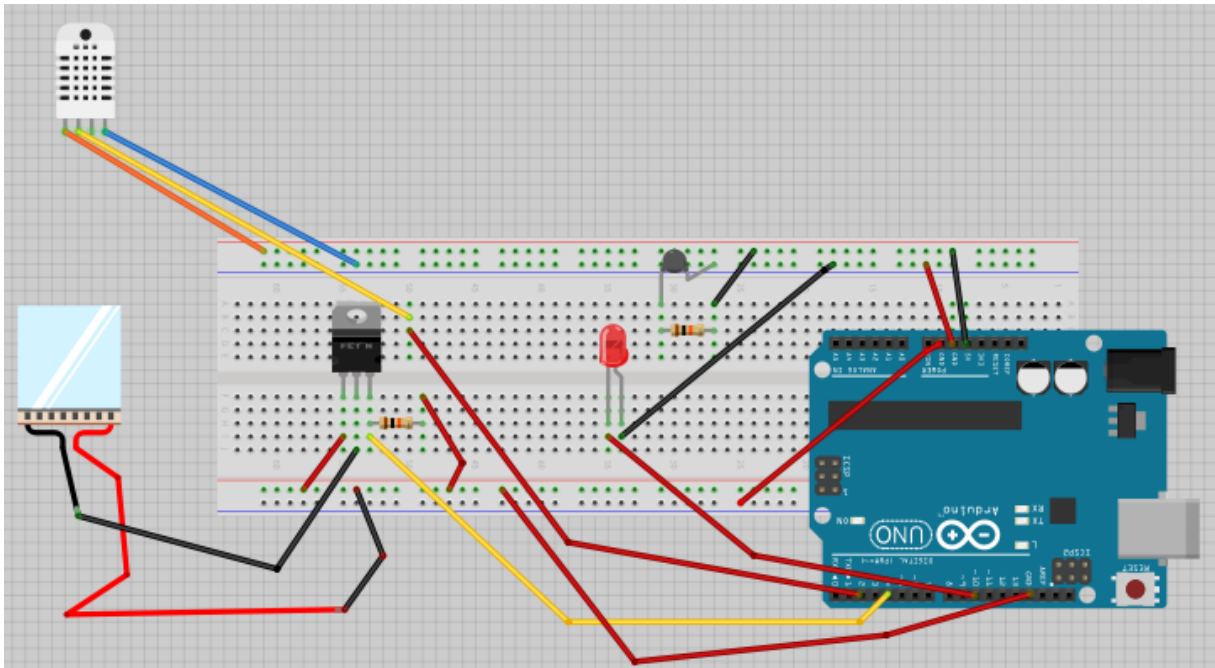


Figure 1 : plan complet du circuit

## Formules :

**-Formule pour calculer le point rosée :** Cette formule a pour but de nous alerter sur la condensation (si la température intérieur dépasse le point rosée la condensation se génère).

Son calcul pour l'air (dans les limites de température usuelles de 0 à 60°C) est donné par la formule empirique de **Magnus-Tetens**.

$$Tr = 237.7 * K / (17.7 - K)$$

Où K est une constante égale à :  $(237.7) * Ta + \text{Log}(\text{Hum})$

Ta (Celsius) = Température ambiante sèche (c'est à dire sans humidité et sans influence additive due aux radiations).

Hum (nombre pourcentage) = humidité relative.

**-Formule pour calculer a température extérieure :** Cette formule a pour but de calculer la température externe via une sonde PT100.

La **relation de Steinhart–Hart** modélise l'évolution de la résistance électrique d'un semi-conducteur selon sa température.

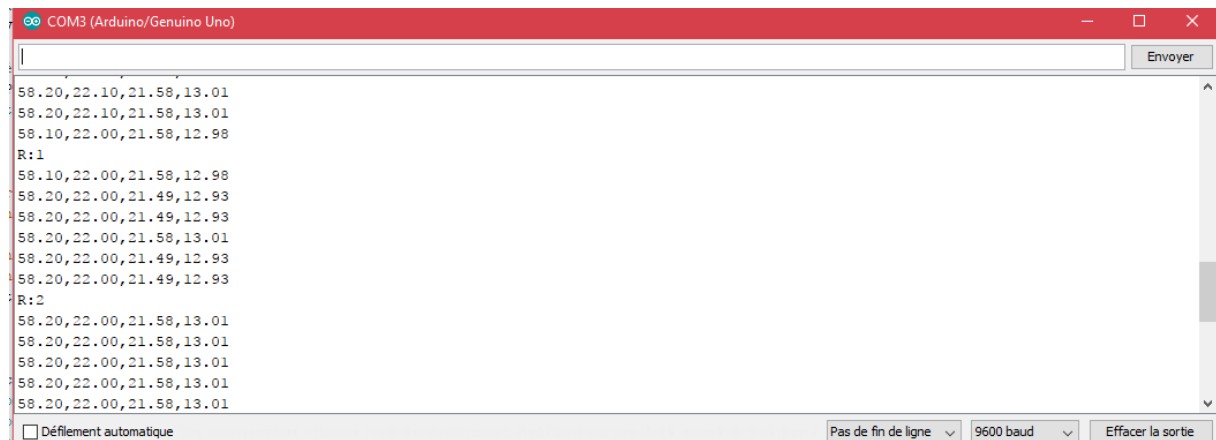
Cette loi peut s'écrire :

$$1/T = A + B * \ln(R) + C * \ln(R)^3$$

T : Température recherchée (Kelvin)

R : Résistance de la sonde

A, B, C : les coefficients de **Steinhart-Hart** qui caractérisent chaque sonde



The screenshot shows the Arduino IDE serial monitor window titled 'COM3 (Arduino/Genuino Uno)'. It displays a list of data points for two probes, R:1 and R:2. Each data point consists of five comma-separated values. For R:1, the values are (58.20, 22.10, 21.58, 13.01), (58.20, 22.10, 21.58, 13.01), and (58.10, 22.00, 21.58, 12.98). For R:2, the values are (58.10, 22.00, 21.58, 12.98), (58.20, 22.00, 21.49, 12.93), (58.20, 22.00, 21.49, 12.93), (58.20, 22.00, 21.58, 13.01), (58.20, 22.00, 21.49, 12.93), and (58.20, 22.00, 21.49, 12.93). The window includes a search bar at the top, an 'Envoyer' button, and a status bar at the bottom with options for 'Défilement automatique', 'Pas de fin de ligne', '9600 baud', and 'Effacer la sortie'.

Figure 2-Affichage console Arduino

## Test de notre projet utilisant ces calculs :

### Calculs du point de rosée avec nos valeurs données :

Nous avons pour une humidité donnée : 42.6, une température externe de 22 °C et un point de rosée de 7.85°C

Prouvons que le point de rosée égale à 7.85 °C ?

Point de rosée =  $(237.7 * (((17.27 * \text{température}) / (237.7 + \text{température})) + \log(h/100))) / (17.27 - (((17.27 * \text{température}) / (237.7 + \text{température})) + \log(h/100)))$

Point de rosée = 8.60 °C

Nous avons une différence de 0.75 °C ce qui extrêmement proche de notre point de rosée calculé.

## Comment ça marche ?

### Matériaux utilisés :

- Un frigo USB
- 2 résistances 10 K  $\Omega$
- Thermistance S861

Ces thermistances miniatures en perles CTN encapsulées série S861 d'Epcos offrent une haute précision de mesure et une réponse thermique rapide. Les thermistances série S861 sont destinées à la détection de températures et à la réalisation de mesures.

Tolérance de résistance  $\pm 5\%$  type J40,  $\pm 1\%$  type F40

- Réponse thermique rapide
- Précision de mesure élevée
- Encapsulation en résine époxy.
- Fils conducteurs en nickel plaqué argent isolés au PTFE
- Puissance dissipée maximale : 60 mW @ 25 °C
- Facteur de perte (dans l'air) : 1,5mW/C (env.)

• Constante de temps thermique (dans l'air) : 15 secondes (env.)

• Taille du corps : 2, 41 Ø x x 6, 5 mm

Grande plage de températures de - 200 °C à 850 °C

Courbe caractéristique quasi linéaire

Précision élevée

Bonne interchangeabilité



#### -Capteur DHT-22

Figure 3-Matériaux utilisés

La DHT22 est un capteur à bas coût permettant d'acquérir une température et une humidité ambiante d'une manière numérique. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer la température et l'humidité de l'air et la transmet d'une manière numérique sur un bus série. Les données sont actualisées toutes les 2 secondes.

La connexion de ce capteur est très simple, il suffit de relier le premier pin à gauche à l'alimentation (3V à 5V), le pin central sur une pin Arduino déclarée en entrée (INPUT) et le pin de droite à la masse (GND).

Ce capteur est semblable au DHT11 mais il dispose d'une plus grande précision et plage de mesure, par contre il est un peu plus cher et un peu plus gros.

Ce capteur est vendu avec une résistance de pull-up entre 4,7KΩ et 10KΩ pour relier le pin DATA au VCC.

#### Caractéristiques

Alimentation : 3 to 5V alimentation et data

Consommation : 2.5mA max pendant la conversion

Etendue de mesure humidité : de 0% à 100% avec une précision à 2-5%

Etendue de mesure température : de -40°C à 80°C avec une précision ±0.5°C

-MOSFET

-LED

-Carte Arduino Uno



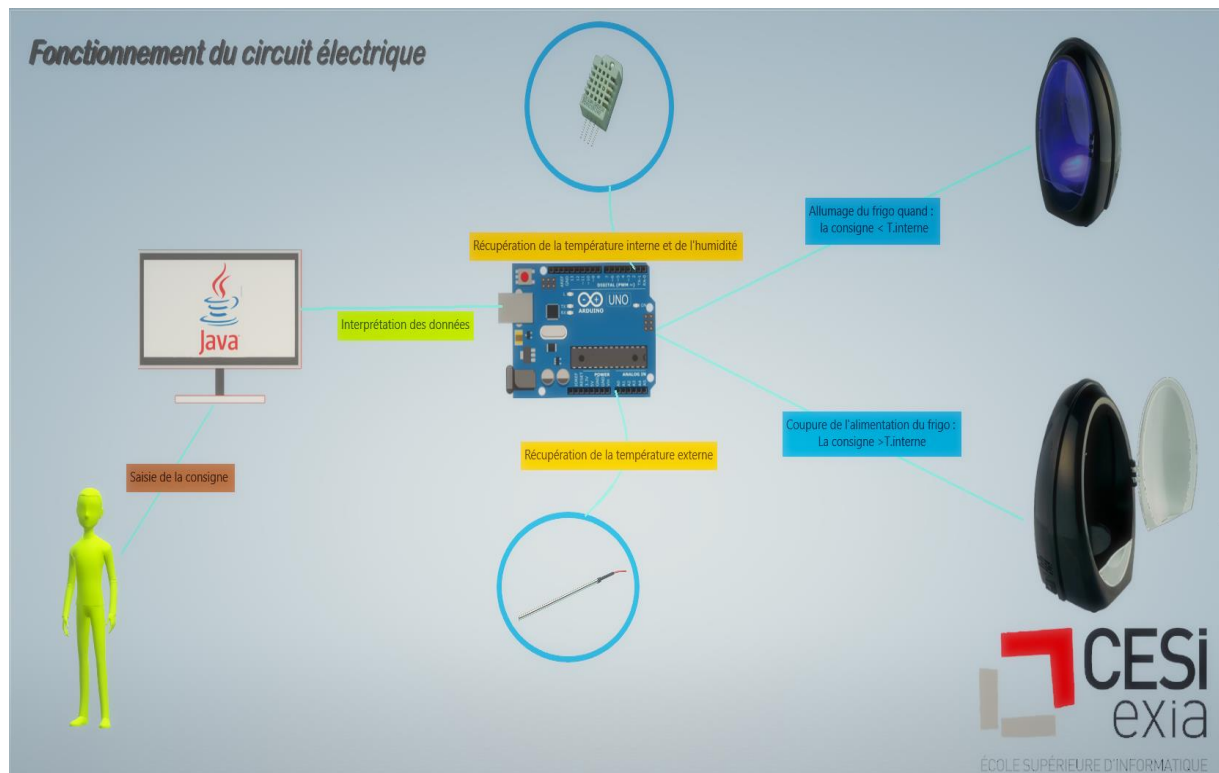


Figure 4-Fonctionnement du circuit électrique

Notre frigo est relié à l'Arduino ainsi qu'à une thermistance qui détecte la température externe et le capteur DHT qui détecte le taux d'humidité et la température à l'intérieur du frigo.

Le défi de ce projet est de créer une interface graphique Java grâce à l'IDE Eclipse qui permet d'afficher les différentes données recueillies par les capteurs et permettre à l'utilisateur de définir une température idéale (Consigne).

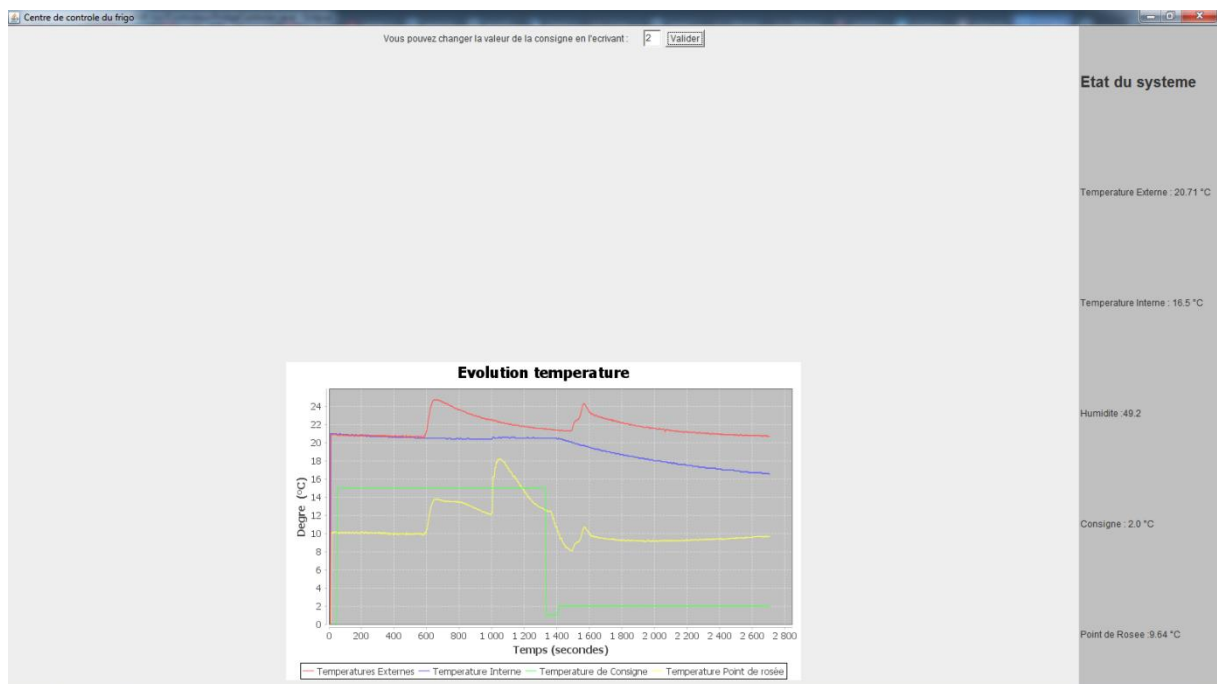


Figure 5- Affichage interface JAVA

Donc notre programme permet de :

- récupérer et d'afficher les températures interne et externe du frigo
- calcule le point de rosée et envoi une alerte de condensation
- donner une température consigne pour garder toujours la même température

### Architecture utilisée (Code JAVA) :

Afin d'avoir un code beaucoup plus souple et maintenable, nous avons utilisé l'architecture MVC (Modèle - Vue - Contrôleur).

- La vue : fournit une interface homme-machine qui permet à l'utilisateur d'interagir avec le frigo en modifiant la consigne (température voulue)
- Le modèle : contient les données dont nous avons besoin (température du réfrigérateur, taux d'humidité...)
- Le contrôleur : ce dernier traite les actions de l'utilisateur, ainsi il modifie les données du modèle et de la vue

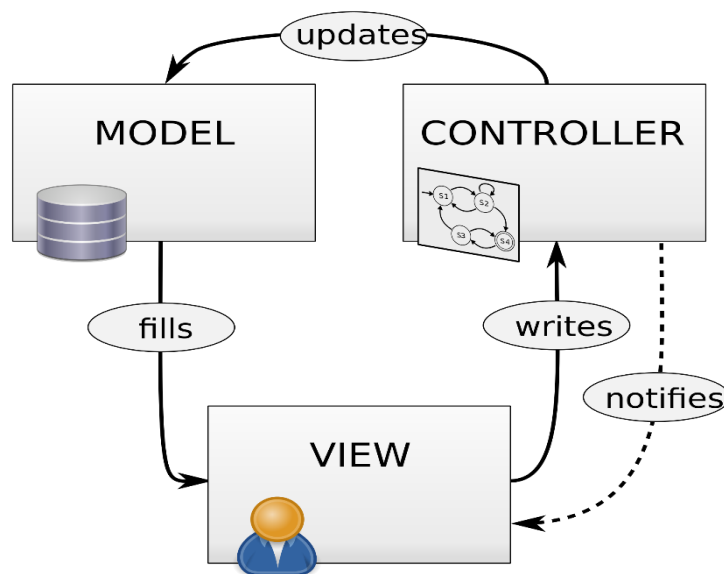


Figure 6 -MVC pattern

## Bilans personnels :

Mehdi Belhadj : ma partie consistait à coordonner le projet tout en s'assurant les tâches sont correctement effectuées par les membres de l'équipe,

Rayan Lakhdar : pour mon premier projet scientifique j'ai eu la chance d'avoir un groupe pas comme les autres aussi drôle que compétant. Je me suis occupé avec Marouane du code java tout en essayant d'acquérir la base de ce langage j'ai aussi bien assimilé les montages et le code Arduino, tout ça en un seul et unique projet qui ne consistait pas qu'à coder ou rester devant ton ordi etc. ... C'était un tout de la science de l'électronique de la physique et de la programmation. Finalement côté environnement de travail l'ambiance était là le travail était là je pense qu'il y a un seul et unique mot qui décrit vraiment notre groupe : efficace.

Mourad Djellouli : pour ma part j'ai trouvé ce premier projet de deuxième année très enrichissant, il nous a permis de mettre en pratique nos notions acquises en Java et en langage Arduino, tout en associant cela au côté scientifique (que j'apprécie tant), très présent en ce début d'année.

Personnellement je me suis principalement occupé du code Arduino ainsi que le montage du circuit électrique avec Mehdi, globalement je suis satisfait du résultat final nous avons pu atteindre nos objectifs dans un délai assez réduit et je tiens à remercier toute l'équipe, qui n'a pas fait les choses à moitié, spécialement Rayan qui a su démontrer une capacité d'adaptation remarquable et un sérieux irréprochable.