

Le projet **Pi-Fridge** a pour but de mettre en commun nos connaissances en physique sur le domaine de la thermodynamique avec celles de l'électronique et de la programmation Arduino et Java.

Nous avons réalisé un **mini-frigo USB** à l'aide d'un module Peltier, de deux capteurs et d'une carte Arduino. En connectant l'Arduino à l'ordinateur il est possible de surveiller l'évolution de la température et de l'humidité du réfrigérateur, mais surtout de pouvoir régler la température souhaitée dans le frigo.

L'interface est développée avec JavaFx pour une grande compatibilité est la grande flexibilité que cela offre.

La **réfrigération** est assurée par un **module Peltier** contrôlé par l'Arduino grâce à un signal analogique envoyé sur un **transistor MOSFET** connecté au module Peltier

Le **capteur de température**, une **thermistance de 10KΩ**, est montée en pont diviseur de tension avec une **résistance de 10KΩ**. La sortie du pont diviseur de tension permet de récupérer une valeur qui sera ensuite traduite en résistance.

Le **capteur d'humidité** utilisé est le **DHT22**, un hygromètre qui va communiquer avec l'Arduino et lui fournir **l'humidité relative** captée avec une précision de plus ou moins 2%.

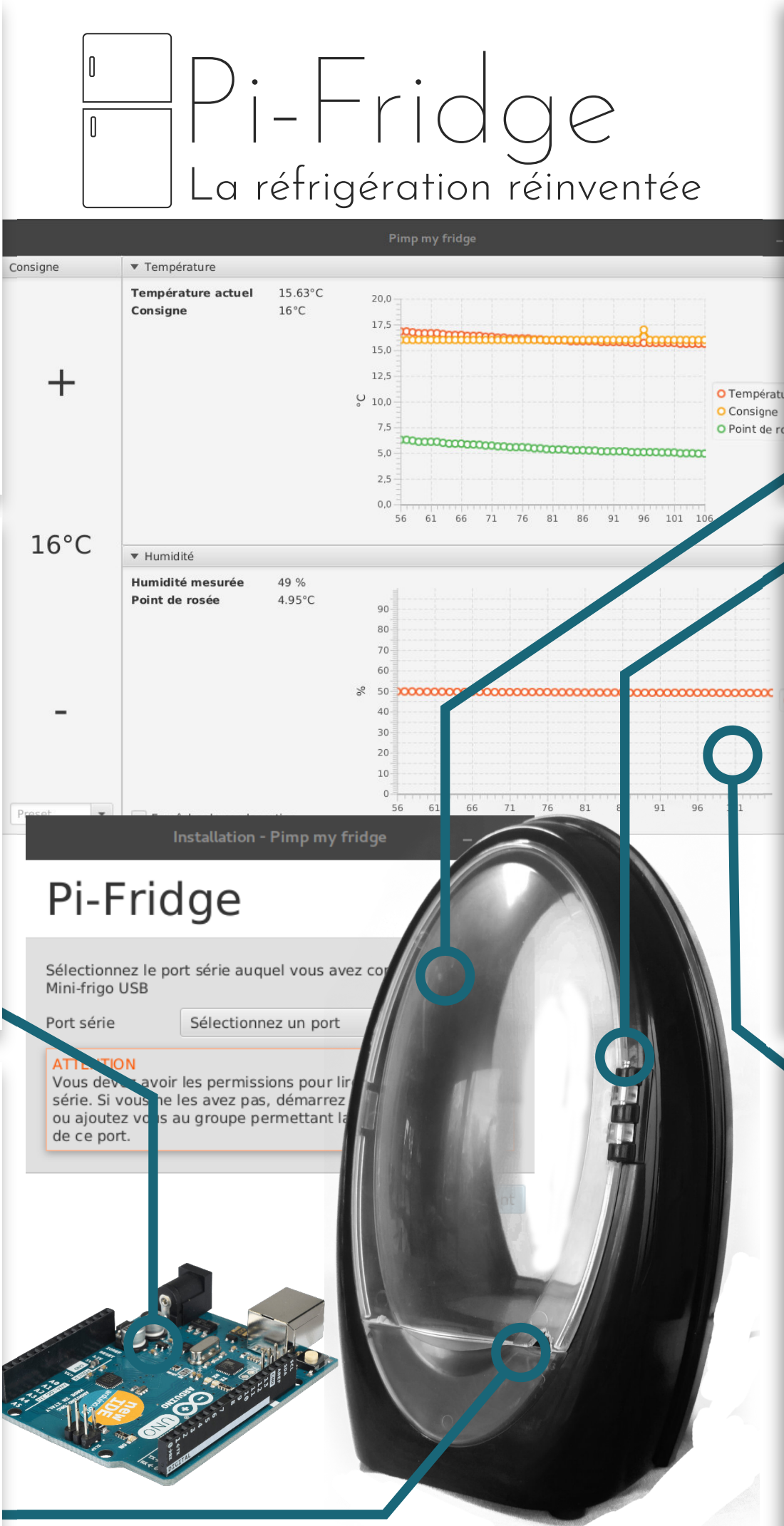
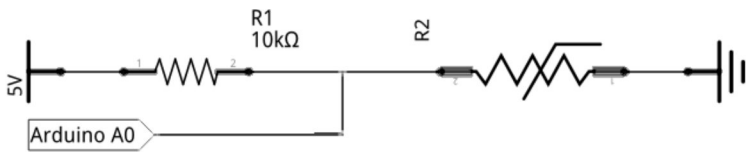
En plus de ces composants, une **LED** est installée afin de signaler que le frigo est en fonctionnement, mais aussi de signaler quand la température souhaitée est atteinte. Un **interrupteur** est aussi fixé au boîtier afin d'allumer le frigo.

Grâce à la **relation de Steinhart-Hart**, nous avons modélisé la température selon l'évolution de sa résistance électrique :

$$\frac{1}{T} = A + B \times \ln(R) + C \times (\ln(R))^3$$

Cette formule nous a permis de mesurer la température dans le système avec une précision de plus ou moins 1%.

La valeur de sortie de la thermistance est traduite en résistance et est insérée dans la formule de Steinhart-Hart afin d'obtenir la température du système.

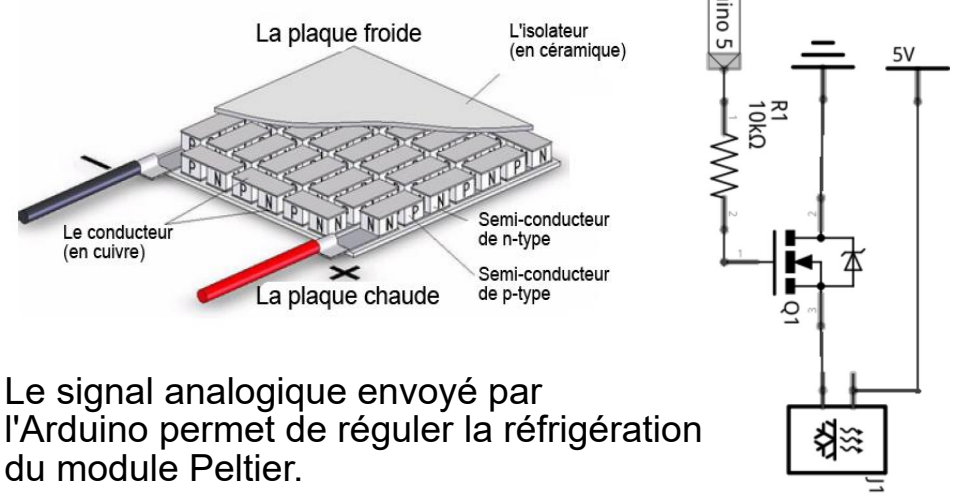


Le **point de rosée** est la **température la plus basse** à laquelle une masse d'air peut être soumise sans qu'il ne se produise une formation d'eau liquide par saturation. Cette température est calculée grâce à la formule suivante :

$$T_R = \frac{237.7 \times K}{17.27 - K} \quad K = \frac{17.27 \times T_a}{237.7 + T_a} + \ln(R)$$

Le point de rosée est calculé grâce aux valeurs renvoyés par les capteurs d'humidité et de température, et le programme va **empêcher** le frigo de **descendre en dessous** de la température de rosée afin d'éviter la **condensation** sur la canette.

Le **module Peltier** permet la réfrigération du système. Il est contrôlé par un **transistor MOSFET** relié à l'Arduino. Il va transformer un **courant électrique** en une **différence de température**. Sur le côté chaud un ventilateur est ajouté afin d'évacuer la chaleur.



Le signal analogique envoyé par l'Arduino permet de réguler la réfrigération du module Peltier.

La communication entre l'Arduino et l'interface client Java se fait par le biais du port série grâce à la bibliothèque **RXTX**. Les données sont parsées en **JSON** selon un modèle.

La **température et l'humidité actuelle** du système sont envoyées à l'interface client, qui à son tour envoie la **consigne** entrée par l'utilisateur. L'Arduino va recevoir la consigne et réguler la température du système pour atteindre cette consigne.

L'utilisateur peut **choisir la température** du frigo et de **visualiser l'évolution de la température** et de l'humidité sur le temps mais aussi limiter la température de consigne pour empêcher la condensation sur la canette.

