

BE Programmation Orientée Objet

Réalisation d'un dispositif de contrôle climatique intelligent

Andrea PEREZ - Léa SCHEER - Christ MBOUMBA

11 Mai 2022

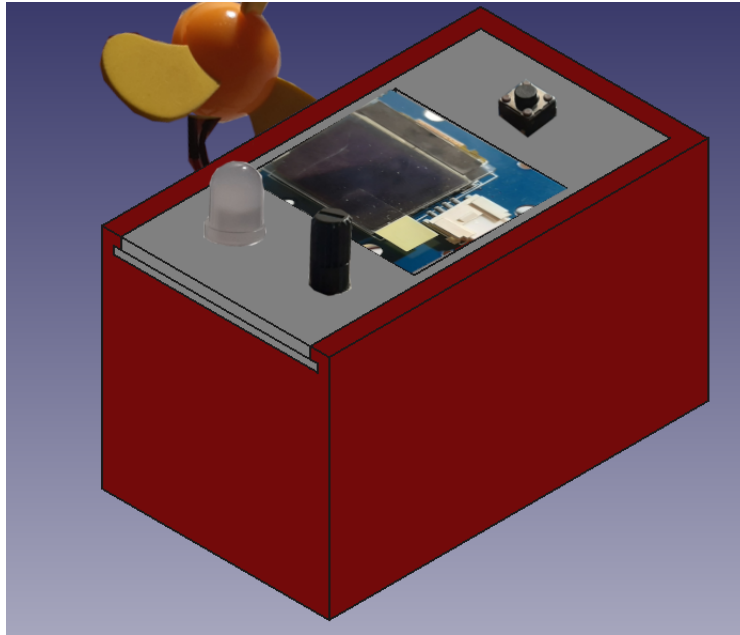


Figure 1: Maquette du ventilateur intelligent iFAN v1.0

1 Présentation générale du système

Dans le cadre de ce Bureau d'Etude de Programmation Orientée Objet, nous avons décidé d'implémenter un système de ventilateur intelligent. Ce dispositif possède deux modes. Au démarrage, l'utilisateur est invité à sélectionner le mode qu'il souhaite : Automatique ou Manuel.

Le mode **Manuel** est le plus basique. S'il est sélectionné, l'utilisateur a seulement la possibilité de modifier la vitesse du ventilateur à l'aide du potentiomètre mis à disposition. Une barre de remplissage s'affiche en fonction de la vitesse. Plus le ventilateur tourne vite, plus elle se remplit. Un bouton "cancel" est également présent si l'on veut retourner au menu initial et arrêter le ventilateur.

Le mode **Automatique** est un peu plus chiadé. Une fois sélectionné, un message demande d'abord si l'on désire activer le mode nuit. Si on sélectionne non alors une fois passé 20h (en admettant que c'est l'heure à laquelle se couche le Soleil) l'interface reviendra au menu initial et le ventilateur s'arrêtera s'il ne l'a pas déjà été. Pour ce faire, le menu suivant consiste à demander l'heure locale à l'utilisateur qui règle alors chaque digit manuellement. Après ce réglage, l'heure est constamment mise à jour et le système stoppe alors la ventilation une fois l'heure limite

atteinte. Par contre, si c'est oui qui est sélectionné alors le ventilateur ne s'arrêtera pas une fois passé 20h et il sera de la responsabilité de l'utilisateur de l'arrêter. Dans ce cas là l'utilisateur n'a pas à régler l'heure.

Une fois la configuration "mode nuit" passée, il est demandé à l'utilisateur de choisir la température qu'il souhaite atteindre. Il peut choisir une température entre 20.0°C et 35.0°C avec un pas de 1.0°C. Une fois validée, le menu suivant récapitule la température souhaitée par l'utilisateur ainsi que la sélection mode nuit ou non. Il y a également le bouton "cancel" pour encore une fois retourner au menu initial et arrêter le ventilateur.

L'implémentation d'un correcteur proportionnel permet de régler la vitesse du ventilateur jusqu'à ce que la température souhaitée soit atteinte. La lampe est un indicateur visuel de la plage de température dans laquelle on se trouve : bleu (moins de 22°C), orange (entre 22°C et 27°C), rouge (jusqu'à 33°C) et rouge clignotant au-delà.

2 Solutions techniques proposées

2.1 Solutions matérielles

Notre projet comporte trois regroupements de périphériques qui sont :

- Interface Homme Machine (IHM) composée d'un écran OLED, un potentiomètre, un bouton poussoir et une LED RGB.
- Un actionneur qui est le ventilateur relié à un pont H pour pouvoir contrôler sa vitesse avec une PWM
- Un capteur de température

Nous avons été amenés à coder sur une Arduino Genuino 101 au lieu d'une ESP82 car nous avons besoin de deux entrées analogiques pour le potentiomètre et le capteur de température.

2.2 Architecture logicielle

Nous avons décidé de créer une classe par regroupement de périphériques. Ainsi nous avons 3 classes qui nous permettent de communiquer et agir à l'extérieur du microcontrôleur. Puis nous avons créé deux autres classes pour gérer la logique de notre programme :

- La classe `Utilisateur` est une classe virtuelle qui permet au programme de garder en mémoire la configuration et le mode choisis par l'utilisateur ainsi que de lancer et contrôler le ventilateur selon les paramètres choisis. Il existe deux classes filles qui héritent de cette classe : la classe `Utilisateur_manuel` et la classe `Utilisateur_automatique`. Elles correspondent aux deux modes possibles de configuration. Chacune redéfinit la méthode virtuelle `lancer` et implémente le fonctionnement que doit avoir le système dans chaque mode.
- La classe `Temps` fournit des méthodes et des opérateurs pour manipuler le temps en heures et minutes à partir d'un `array` de la librairie `std` où des entiers longs représentent des secondes. Nous utilisons les méthodes de la classe pour accéder aux valeurs et informations du tableau. Grâce à la surcharge d'opérateurs, le passage entre types est beaucoup plus simple.

Les trois classes qui nous permettent d'interagir avec l'extérieur sont :

- Classe `IHM` avec laquelle on affiche des informations sur l'OLED et on récupère les choix de l'utilisateur grâce au potentiomètre et bouton poussoir. Elle permet aussi de contrôler la couleur de la LED.
- Classe `Ventilateur` permet de configurer la sortie D6 en PWM afin de contrôler sa vitesse.
- Classe `Capteur_temp` qui permet de lire sur l'entrée analogique A1 la valeur de la température.

Nous avons aussi un fichier `.h` appelé `definitions.h` où sont définis les deux types énum que nous utilisons :

- `mode_utilisation` qui peut être automatique ou manuel
- `mode_nuit` qui peut prendre activé ou pas

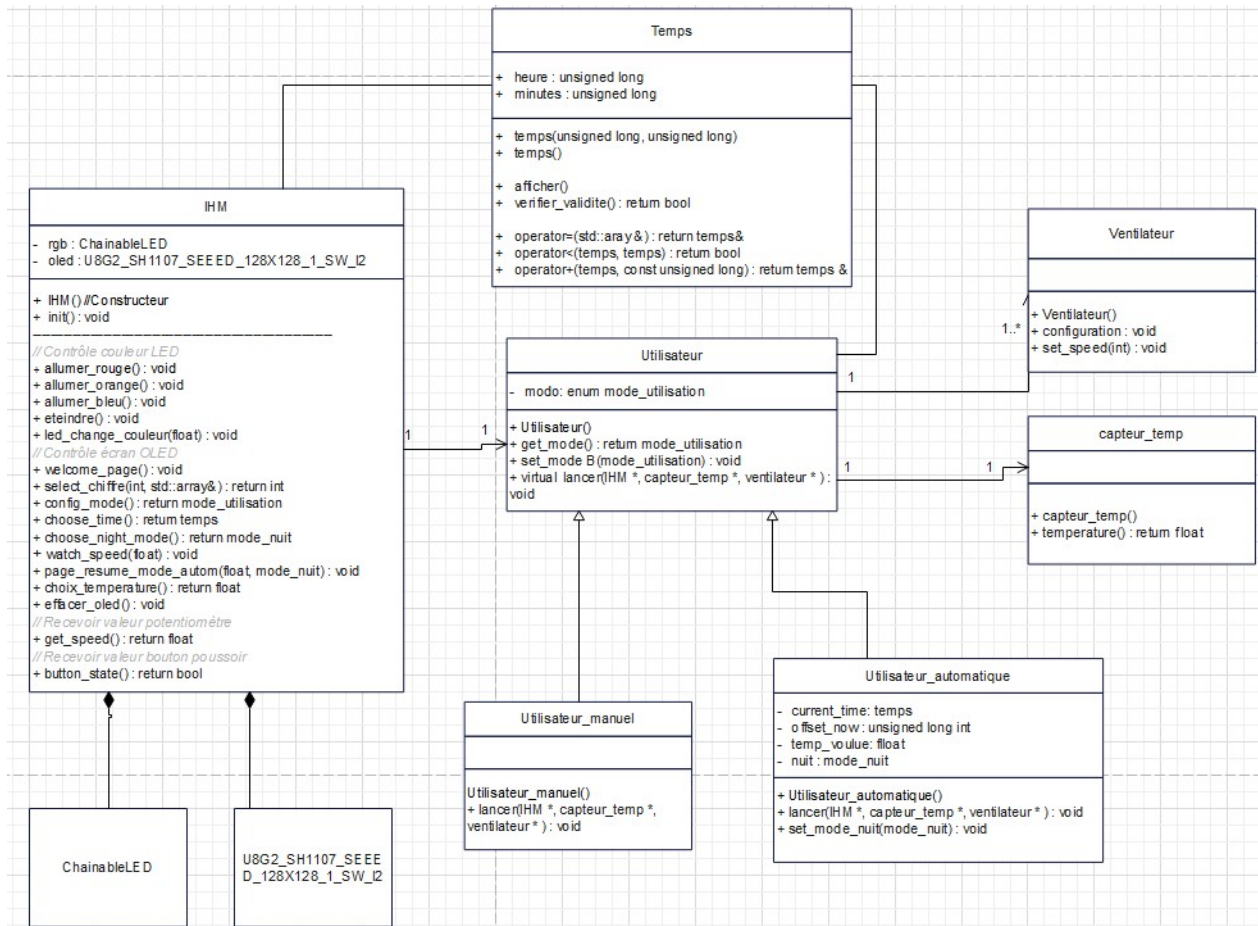


Figure 2: Diagramme de classe du projet

3 Conclusion

Ce Bureau d'Etude s'est révélé très formateur et ludique. D'une part, nous avons pu utiliser les connaissances déjà acquises lors des séances de TD et les mettre au profit au sein d'un projet plus global conçu de A à Z par nos soins. Si la tâche paraissait fastidieuse au départ, les idées se sont toutefois mises en place petit à petit dans nos esprits et ont commencé à prendre forme rapidement en se matérialisant en code. Initialement, bien que pas très "graves", nous avons beaucoup d'erreurs lorsque nous codions (problèmes de compatibilité de carte Arduino, correspondance de pins, etc.). Au fur et à mesure, nous avons su trouver les solutions pour chacune d'entre elles. Une fois cette étape franchie, nous étions lancées et le projet a avancé très vite. C'est un projet qui nous a demandé du temps et beaucoup de concentration mais qui a été enrichissant et valorisant au vu des efforts fournis. Toutes les notions fondamentales vues en TD ont été acquises ou du moins bien comprises; certaines subtilités que nous n'avions pas forcément avant ont de plus été assimilées. L'aspect le plus "délicat" a sûrement été l'intégration de toutes les classes ensemble, mais là encore nous avons persévéré et abouti au résultat attendu. Nous sommes donc globalement très satisfaites quant à la réalisation de ce projet qui de plus nous paraît pertinent d'un point de vue "utilité".