Tom JULLIEN & Thomas BOURSAC 4 AE SE 4

Rapport de projet de C++

1. Introduction

Dans ce projet, nous avons choisi de créer une station météorologique afin de mettre en avant les différentes notions que nous avons apprises sur le langage C++. Pour cela nous avons utilisé différents capteurs nous permettant de retransmettre l'humidité, la température ainsi que la météo en temps réel.

2. Mise en forme de l'idée

Afin de rendre le projet plus atypique, nous avons eu l'idée de retranscrire la météo en temps réel sur un servomoteur avec une roue indiquant les différentes météo possibles (Pluie, nuages, légèrement nuageux, soleil). Pour cela, nous nous sommes basés sur la récupération de la pression atmosphérique puis sur l'association de cette donnée a un type de météo pour enfin calculer un angle proportionnel.
La météo étant en grande partie causée par les variations de pression atmosphérique, nous avons configuré un capteur de pression atmosphérique pour le rendre efficace pour tous les endroits étant au niveau de la mer. Nous nous sommes basés sur une pression de 100 000 hpa. Ainsi, une pression basse (< 100 000 hpa) correspond à un mauvais temps et une pression haute (> 100 000 hpa) correspond à un beau temps (cf Figure 1). Notre programme traduit proportionnellement cette pression et retourne l'angle correspondant sur la roue. Le code permettant de transformer la pression en angle est disponible dans le fichier algo.cpp.ino.

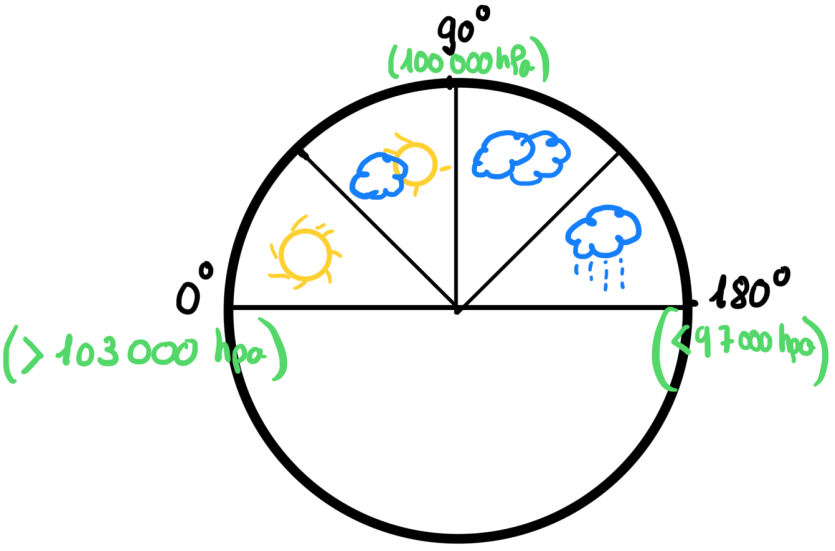

Figure 1: schéma du disque du servomoteur

3. Adaptation vers des classes de C++.

Il fallait à présent organiser notre projet en classes et en objets. Pour cela, nous avons fait le choix de représenter d'un côté tous les capteurs dans une même classe mère appelée "Capteur" et tous les composants servant à afficher sous une classe mère "Affiche". Parmis les capteurs nous possédions :

- un capteur CO2 (pas utilisé par la suite).
- un capteur de température et de pression, le BHP.
- un capteur de température et d'humidité, le DHT.

Parmis les affichages nous avions principalements deux classes filles :

- un écran LCD.
- un servomoteur.

Nous avons ensuite pu établir un diagramme de classe à l'aide des différentes classes énoncées ci-dessus ainsi que leurs attributs :

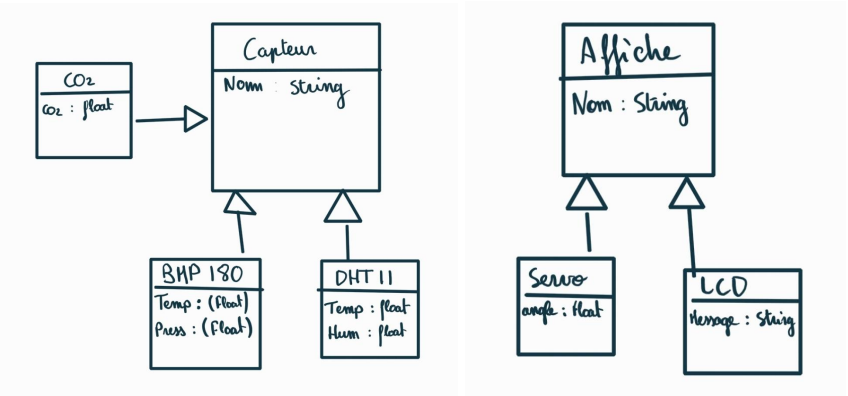

Figure 2: Schéma du diagramme de classe des capteurs et des affichages

4. Fonctionnement du programme et diagramme de séquence

Dans un premier temps, nous actualisons tous nos capteurs, nos affichages ainsi que les différents objets que nous allons utiliser tout au long du programme.

Nous initialisons ensuite une liste provenant de la STL afin de garder en mémoire les différentes températures que nous captons pour effectuer un suivi et une température moyenne tout au long de la journée.

Après avoir initialisé les différents capteurs, nous récupérons les données de ceux-ci tout au long du programme dans les objets à l'aide de fonctions internes aux classes.

Nous avons eu l'idée de surcharger un opérateur afin d'obtenir une température moyenne car nous possédons deux différents capteurs de température. Pour cela nous avons initialisé une classe Température avec un objet pour chaque capteur et un troisième recevant la moyenne des deux autres.

Nous effectuons ensuite le calcul de l'angle à fournir au servomoteur à partir de la récupération de la pression atmosphérique. Nous affichons de façon alternée sur l'écran LCD température et humidité. Enfin, nous mettons à jour la liste STL avec la température.

5. Améliorations possibles et difficultés rencontrées.

Lors de ce projet, nous avons dû nous familiariser avec l'environnement Arduino, nous avons utilisé un Arduino Uno afin de pouvoir utiliser le 5V pas disponible sur ESP82. La gestion des bibliothèques et des différents imprévus tel que la gestion des exceptions impossible sur l'IDE Arduino nous a fait perdre beaucoup de temps sur du débuggage.

Sur l'amélioration de notre projet, nous pourrions inclure d'avantages de capteurs afin d'affiner le résultat qui pour le moment est assez simplifié car ne tenant pas compte d'autres paramètres que la pression atmosphérique ni même l'évolution au cours du temps des données reçues par le capteur. De plus, nous avions pour but d'inclure un 2e servomoteur indiquant la qualité de l'air mais nous n'avons pas trouvé de capteurs adaptés à ce type de données. Enfin, nous avons implémenté un mécanisme d'exception qui, lorsque la température augmente trop, génère une exception changeant la couleur du LCD et affichant "TROP CHAUD, la température dépasse 30°". Cependant, l'IDE Arduino ne prend pas en charge les exceptions. Nous avons cherché sur internet et nous avons trouvé une solution dans laquelle il faut modifier un fichier source de Arduino, nous avons essayé sans succès, le code de l'exception est donc en commentaire dans le code.

6. Conclusion

Ce projet nous aura permis de nous familiariser avec l'environnement Arduino et les différents capteurs tout en mettant en application nos connaissances en C++. Malgré les délais courts et la gestion des imprévus, nous avons pris du plaisir à laisser cours à nos envies. Le projet reste un projet déjà vu mais nous aimons bien son rendu original avec le servomoteur. Cela fut plaisant de se familiariser avec les capteurs, créer un projet de sa conception à la réalisation, de coder et tester nos programmes.