

Lors de cette séance, j'ai tout d'abord cherché à résoudre le problème rencontré avec le capteur de température et d'humidité lors de la séance précédente (un message d'erreur de compatibilité apparaissait lors du téléversement).

En effet, j'ai essayé d'installer une autre librairie plus ancienne dont la version datait de 2015.

Le programme s'est alors exécuté correctement, tout était fonctionnel.

J'ai ensuite passé le relais, sur cette partie du projet, à mon binôme.

Je me suis consacré ensuite au problème qui faisait que les moteurs ne tournaient pas à de faibles vitesses pour les stores.

Pour ce faire monsieur MASSON m'a conseillé de modifier mon code en ajoutant des « delay » afin de faire tourner les moteurs à des vitesses importantes tout en contrôlant la montée des stores.

Voici un morceau du code que j'ai réalisé :

Tout d'abord j'ai créé une fonction nommée MOTEUR afin d'alléger la taille de mon code :

```
void MOTEUR() {  
    analogWrite(ENA, 200);  
    analogWrite(ENB, 200);  
    delay(100);  
    analogWrite(ENA, 0);  
    analogWrite(ENB, 0);  
    delay(100);  
}
```

Le principe est de faire tourner les moteurs à une vitesse importante (200) puis insérer un delay de 100ms avant de les faire s'arrêter. Un autre delay intervient à la fin de la fonction.

J'insère ensuite cette fonction à deux reprises dans le cas où l'utilisateur appuie sur le bouton pour faire monter les store :

```
if (Data=='A') {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    MOTEUR();  
    MOTEUR();  
}
```

Cela permet de faire tourner les moteurs au « coup par coup », à une vitesse importante certes mais marquée par des temps d'arrêts qui permettent une montée ou une descente des stores à une vitesse adaptée.

Dans le code, la fonction MOTEUR n'apparaît que 2 fois car c'est le nombre de fois suffisant pour que les stores descendent ou remontent totalement.

Cette solution m'a permis ensuite de me consacrer à une autre partie du projet avec mon binôme.

En effet sur la partie du projet concernant le capteur de température, notre objectif est de pouvoir afficher la température et l'humidité sur un écran à l'intérieur de la maison en gérant l'affichage de l'un ou de l'autre par bluetooth sur le téléphone.

Nous avons donc du réaliser un trou sur la maquette de la maison afin d'y installer un second écran (le premier servant au système de déverrouillage de la porte avec le digicode).

Nous avons donc dans un premier temps pris les dimensions de l'écran puis tracé sur la maquette un rectangle adapté :



Nous avons commencé à réaliser le trou à la perceuse, mais nous nous sommes vite rendu compte que cet outil n'était pas l'outil idéal pour réaliser ce trou. Nous nous sommes procuré un autre outil beaucoup plus efficace pour le réaliser :



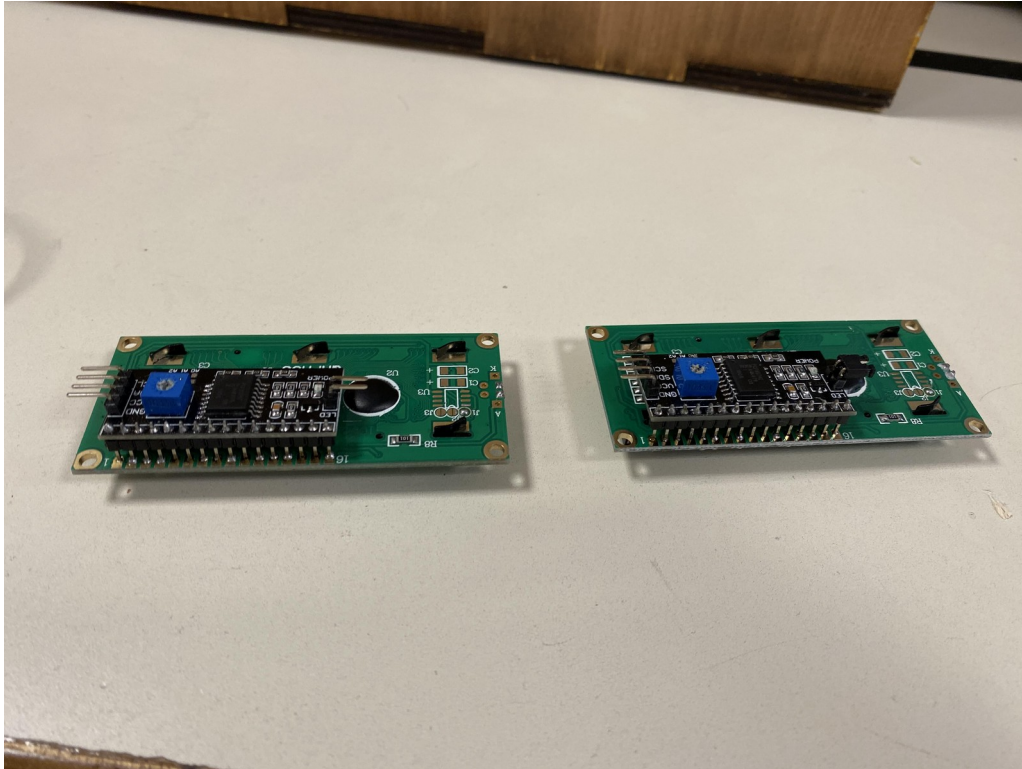
Ce travail était tout de même fastidieux car il fallait réaliser la percé avec beaucoup de soin pour ne pas dépasser les contours du dessin établi, d'autant plus que l'outil en photo ci-dessus était très puissant. Nous avons donc obtenu le résultat ci-dessous, adapté à la taille de l'écran qui s'y emboîte correctement :



Nous avons ensuite limé les contours du cadre pour que ceux-ci soient plus propres.



Enfin nous avons profité de la fin de séance pour nous procurer un second prototype noir a souder sur l'écran qui nous permet de simplifier les branchements de l'écran (je ne me souviens plus de son nom voir photo ci-dessous) :



Cependant, le professeur nous a indiqué que l'adresse de ces deux prototypes étaient par défaut les mêmes et lorsque nous essaierons de connecter les deux écran à une même carte Arduino, la carte ne sera pas en mesure d'effectuer une différence entre ces deux ports.

Pour cela monsieur MASSON nous a aidé à modifier l'adresse d'un des deux composants. L'un possède dorénavant une adresse numérotée 26 et l'autre 27.

Nous avons ensuite soudé le composant sur l'écran et vérifié que ce dernier était fonctionnel. Nous étions alors très satisfait.