

## **Compte rendu Séance 8 + Travail personnel – Projet Arduino – Simulateur de course automobile**

- **Introduction**

Cette semaine est la dernière avant le rendu final du projet. Nous devons terminer les différentes parties de notre projet pour être sûr que celui-ci fonctionne parfaitement le jour de l'examen. L'objectif de notre séance est de fixer toutes les pièces pour obtenir le résultat final. Pour y arriver, j'ai repris les soudures des extensions pour les boutons du volant. Nous avons également testé tous les boutons poussoirs indépendamment. Enfin, nous avons fixé les boutons au volant et brancher tous les câbles pour tester toutes les pièces du volant.

- **Extension des câbles des boutons**

Lors de la dernière séance, je m'étais trompé sur les dimensions des câbles des boutons poussoirs actifs. Durant mon temps libre, j'ai rajouté des extensions à tous les câbles pour que leur longueur permet à la fois de les fixer à la carte Arduino et que le volant puisse tourner à 180 degrés de chaque côté.

Lorsque nous sommes arrivés en TD, nous avons testé tous les boutons pour savoir si les soudures sont bien faites.

Pour les brancher, nous décidons d'utiliser des boutons poussoirs sans résistance : la commande `pinMode()` peut utiliser le mode de fonctionnement `INPUT_PULLUP`. Une résistance de quelques  $k\Omega$  est connectée à la broche, permettant aux boutons d'être connectés à la carte Arduino sans résistance.

En testant les 6 boutons avec la commande ci-dessus, ils fonctionnent parfaitement.

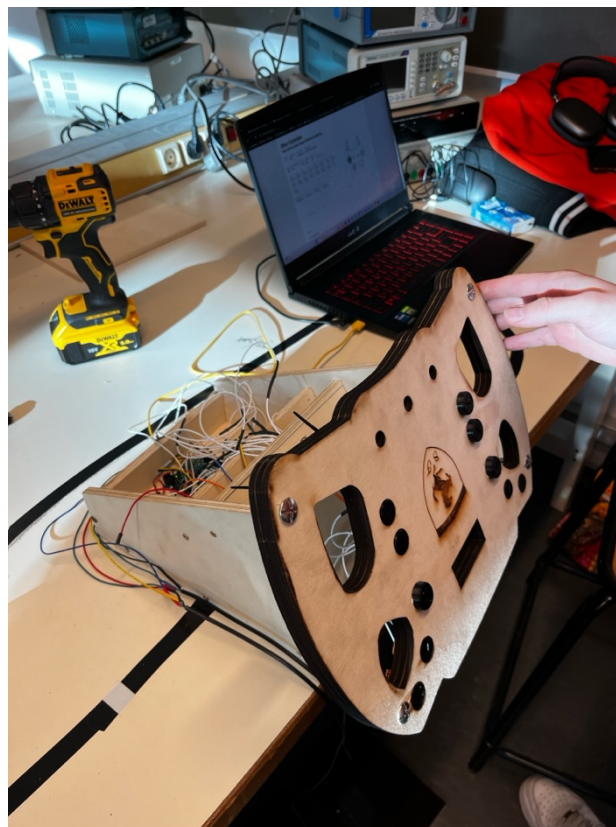
Durant la séance, on me fait remarquer que les soudures effectuées ont quelques défauts malgré leur bon fonctionnement. Je n'avais pas anticipé la taille du dénudage des câbles. En les dénudant trop, les câbles peuvent se toucher et créer un faux contact entre eux. Les conséquences peuvent être lourdes puisque la carte peut, au mieux, ne plus fonctionner et au pire, brûler. Deux solutions se sont offertes à moi : ressouder l'entièreté des câbles avec les boutons poussoirs ou ajouter de la gaine thermo rétractable sur les bouts les plus exposés. Je choisis la seconde solution par manque de temps mais à l'avenir, je ferai plus attention à mes soudures pour éviter ce type de problème.

- **Fixation des pièces actives et passives au volant**

Pour fixer les plaques de soudure au volant, nous les collons à l'aide d'un pistolet. Sur du bois, cette solution nous semble être la plus intéressante. En effet, les plaques ne doivent pas se détacher malgré la pression exercée sur le bouton. Sur le bouton poussoir, nous fixons, d'abord à l'aide du pistolet à colle puis avec de la pâte à fixer, les boutons imprimés en 3D. Le pistolet à colle n'est pas forcément la bonne solution puisque la surface du bouton poussoir est lisse. Pour résoudre ce problème, nous aurions pu limer la surface pour qu'elle accroche mieux au bouton. Finalement, nous décidons d'utiliser de la pâte à fixer.

Pour les pièces passives, nous avons également utilisé de la pâte à fixer. L'avantage de la pâte à fixer est qu'elle est réutilisable et facilement détachable des pièces. En effet, lorsque nous aurons fini le projet, nous devons redonner toutes les pièces utilisées dans le projet. C'est donc un gain de temps pour la fin du projet et nous pourrons ré-utiliser les boutons et la pâte à fixer pour de futurs projets.

Ainsi, voici le résultat du volant avec tous les boutons fixés :



*Volant accroché au support. Téléversement du programme sur la carte Arduino Leonardo*

- **Code utilisé pour le projet**

Durant toute la durée du projet, nous avons testé les différentes pièces du projet indépendamment, au fur et à mesure de notre avancée. En s'assurant du bon fonctionnement au moment opportun, nous réussissons à associer tous les codes pour toutes les pièces sans rencontrer de problème en quelques minutes.

Voici une partie du code utilisée pour le simulateur automobile :

```
void loop() {
  // Read pin states
  boolean pressA = !digitalRead(Pin_ButtonA);
  boolean pressB = !digitalRead(Pin_ButtonB);
  boolean pressX = !digitalRead(Pin_ButtonX);
  boolean pressY = !digitalRead(Pin_ButtonY);
  boolean pressLEFT = !digitalRead(Pin_ButtonLEFT);
  boolean pressRIGHT = !digitalRead(Pin_ButtonRIGHT);

  int pressTrigger = analogRead(Pin_TriggerR);
  int pressTriggerL = analogRead(Pin_TriggerL);
  int joystickValue = analogRead(Pin_Joystick);

  // Lecture des signaux du KY-040 arrivant sur l'arduino
  int etatActuelDeLaLigneCLK = digitalRead(SignalCLK);
  int etatActuelDeLaLigneDT = digitalRead(SignalDT);

  if(etatActuelDeLaLigneCLK != etatPrecedentLigneCLK) {

    // On mémorise cet état, pour éviter les doublons
    etatPrecedentLigneCLK = etatActuelDeLaLigneCLK;

    if(etatActuelDeLaLigneCLK == LOW) {

      // On compare le niveau de la ligne CLK avec celui de la ligne DT
      // -----
      // Nota : - si CLK est différent de DT, alors cela veut dire que nous avons tourné l'encodeur dans le sens horaire
      //         - si CLK est égal à DT, alors cela veut dire que nous avons tourné l'encodeur dans le sens anti-horaire
    }
  }
}
```

*Code Arduino pour le simulateur de course*

- **Test du simulateur automobile sur un jeu**



*Simulateur automobile en action*

Pour relier les boutons poussoirs à la carte Arduino, nous séparons les fils des entrées numériques de ceux pour la masse. Pour ne pas se tromper lors du branchement, nous mettons du serre-câble pour les distinguer.

De plus, pour les faire passer dans le support nous perçons 2 trous distincts pour différencier ceux qui sont branchés à la terre et ceux des entrées numériques.

Ayant le code fonctionnel et toutes les pièces fixées, nous testons notre simulateur automobile sur un jeu vidéo. Toutes les pièces fonctionnent parfaitement et les comportements des pièces sont ceux auxquelles nous nous attendions.

Ainsi, le test se déroule sans accroc et nous permet de jouer comme un pilote !

- **Conclusion**

Durant cette dernière semaine de projet, nous avons terminé toutes les tâches que nous nous sommes fixés dans notre cahier des charges. Nous sommes extrêmement satisfaits de cette séance car tous les efforts fournis durant ces semaines ont porté leur fruit et permettent de faire fonctionner notre simulateur. De plus, nous terminons dans les temps impartis ce qui montre que nous avons bien respecté les délais imposés et que nous nous sommes bien réparti les tâches.

Si nous avons le temps, durant la semaine du 04/03, nous devons effectuer un « cable management ». En effet, dans nos futurs métiers, après avoir conçu un produit, en particulier dans l'électronique, celui-ci doit être entretenu. La maintenance d'une machine est bien plus simple sur une machine bien conçue (dans le sens de la gestion des câbles) que sur une machine mal organisée. De plus, j'aimerais savoir s'il est possible que les pédales de freinage et d'accélération soient considérées comme des potentiomètres car dans le code actuel, celles-ci sont considérées comme des boutons poussoirs.