

Le Air-Duino

Mars 2024

Par BAILET Pauliet & GAROUX Ewan

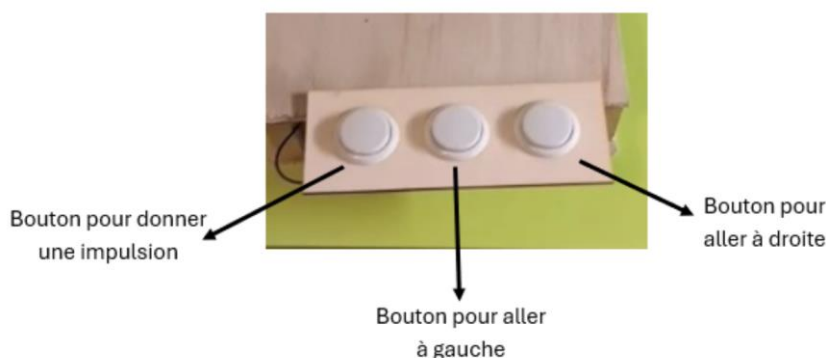


SOMMAIRE:

- Objectif
- Coût du projet
- Matériel utilisé
- Planning initial et final
- Schéma électrique
- Algorithme de fonctionnement
- Problèmes rencontrés
- Bibliographie

L'objectif principal de notre projet est la conception d'un jeu de air-hockey commandé à distance, baptisé le Air-Duino. Ce système s'appuiera sur deux rails permettant le déplacement de pièces mobiles équipées de solénoïdes. Tout comme dans la version traditionnelle du jeu, l'objectif d'Air-Duino est de propulser le palet dans le but adverse pour marquer des points.

Le principe du jeu est intuitif : les joueurs doivent manœuvrer le gardien de but d'un côté à l'autre des cages pour bloquer le palet et le dévier de l'autre côté du terrain. Pour interagir avec le jeu, les participants disposent de boîtiers de contrôle situés de part et d'autre de l'aire de jeu, chacune dotée de trois boutons :



Pour rendre notre projet plus abordable, nous l'avons divisé en trois parties:

1. Le déplacement du goal. Le goal se déplace sur un rail le long des cages.
2. L'impulsion du palet. Pour que le palet puisse traverser le plateau de jeu, il faut lui donner une certaine impulsion. Nous avons trouvé le moyen de le faire grâce à un solénoïde fixé sur le goal.

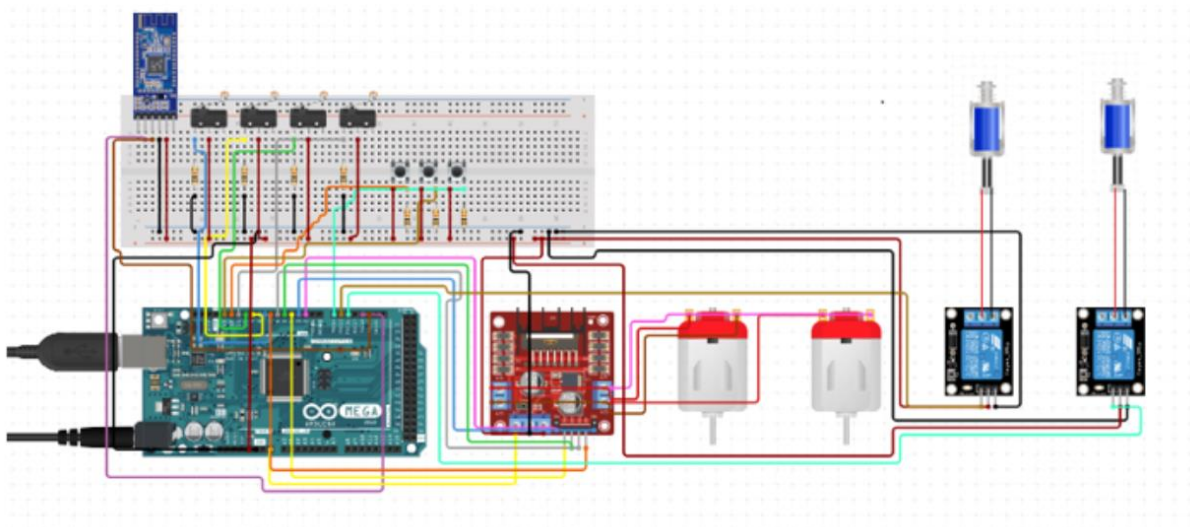
Sur le tableau on voit en rouge le travail effectué par Ewan et on rose le travail effectué par Pauline.

Nous constatons que des parties du projet telles que l'axe de déplacement ou les solénoïdes nous ont pris plus de temps que prévu. De plus, en dehors des heures de cours nous avons travaillé près de 26 heures.

TOTAL DES HEURES TRAVAILLÉES: 50 HEURES

Pour un ingénieur payé annuellement 38k euros pour 1600h de travail, ici il sera payé 1187.5€.

Voici le schéma électrique général de notre projet :



De façon plus détaillée voici les trois parties de notre projet:

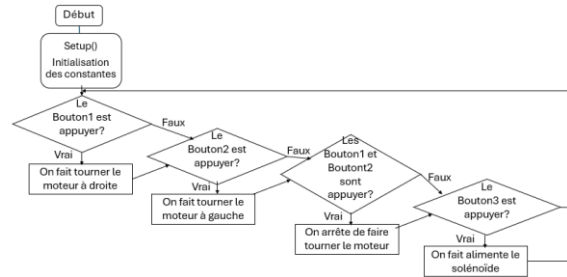
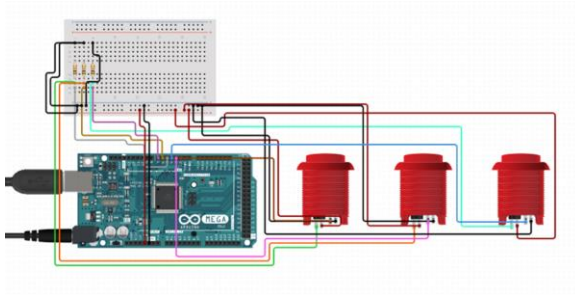
- Pour la boîte à boutons :

Notre jeu est composé de trois boutons, comme expliqué dans les règles du jeu, chaque bouton effectue une action différente.

En appuyant sur le bouton de droite, on appelle la fonction tourne à droite et le moteur est activé du côté droit.

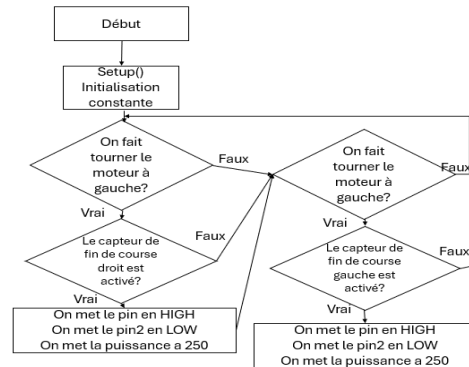
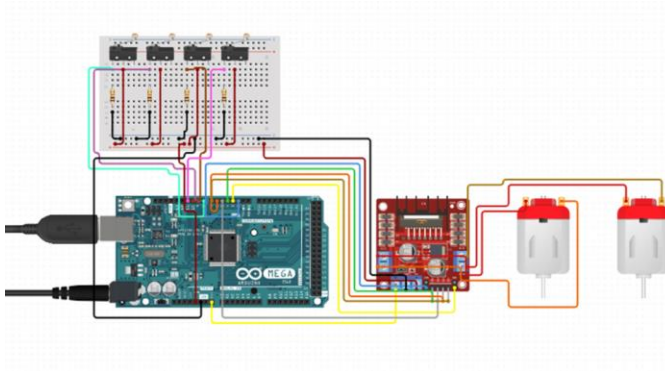
De la même manière, en appuyant sur le bouton de gauche, le code appelle la fonction tourne à gauche et le moteur s'active du côté gauche.

Enfin, en appuyant sur le dernier bouton, le relais reçoit 5V et le solénoïde s'active.



- Pour le fonctionnement des moteurs :

Le code du moteur est composé des fonctions tourne à gauche et tourne à droite. Celles-ci sont appelées en appuyant sur leur bouton associé à condition que les capteurs de fin de courses ne soient pas activés, sinon le moteur s'arrête.

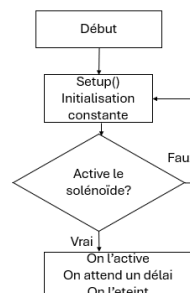
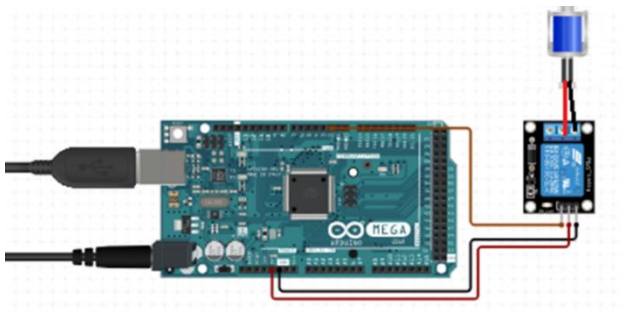


- Pour le solénoïde :

Le solénoïde est relié à un relais, celui-ci permet d'envoyer 5V ou 0V en fonction de la borne sur laquelle il est branché.

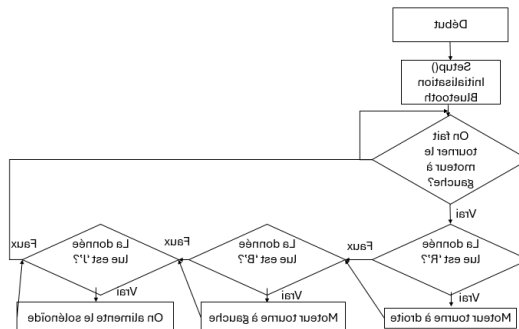
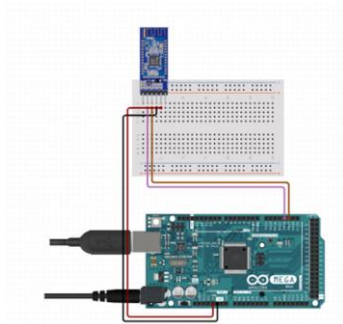
De plus le solénoïde est branché sur une alimentation externe de 12V.

Dans le code du solénoïde nous avons ajouté une fonction delay(), elle sert de condition au joueur qui ne pourra pas appuyer plusieurs fois d'affilées sur le bouton du solénoïde.



- Le module Bluetooth :

Pour cette partie on a simplement connecter un module bluetooth et recuperer les donnés qui ont été transmises par la telecommande (dans ce cas le telephone avec l'application bluetooth electronics) puis en fonction de l'information reçu on réalisé les differentes actions.



Les problèmes rencontrés:

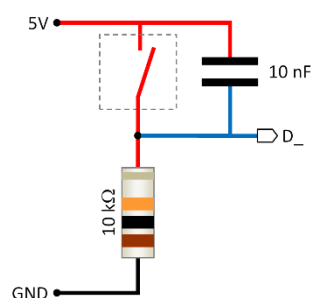
Lors de notre projet nous avons rencontré trois problèmes majeurs.

Le premier était un problème d'effet rebond sur les boutons poussoirs.

En ce qui concerne l'effet de rebond. Lorsqu'un interrupteur change d'état, il y a souvent des oscillations pendant la période transitoire pour des raisons physiques (mécanique, temps de réponse, ...). Il faut donc laisser un temps suffisant pour que l'état puisse se stabiliser. Cela peut se faire soit par filtrage électronique ou mécanique soit numériquement (avec un programme qui laisse un délai).

Dans notre cas en voulant optimiser le code et réduire les temps de délai, nous avons décidé d'utiliser la solution par filtrage électronique. Pour atténuer le phénomène, nous avons utilisé un condensateur en parallèle avec le bouton. Ce composant sert ici "d'amortisseur" qui absorbera les rebonds. Le condensateur, initialement chargé, va se décharger lors de l'appui sur le bouton. S'il y a des rebonds, ils seront encaissés par le condensateur durant cette décharge. Il se passera le phénomène inverse (charge du condensateur) lors du relâchement du bouton.

Le schéma final du bouton donne donc:



Le deuxième problème rencontré est un problème d'engrenage. Il s'agit ici de la pièce qui fait le contact entre l'axe (tige fileté) et le moteur. Dans un premier temps nous avons décidé de modéliser l'engrenage en 3D et de le coller sur l'axe de moteur. En revanche, cette solution n'était pas idéale car la pièce finissait toujours par s'user. On a donc décidé de chercher des engrenages mécaniques qui puissent durer.

Par manque de temps, le délai de livraison était trop court. Nous avons donc démonté des voitures télécommandées. Comme nous l'espérions, dans les réducteurs nous avons trouvé des engrenages en plastique renforcé. Après plusieurs tests la nouvelle pièce a tenu et nous avons donc décidé de la garder.

Enfin, le dernier problème rencontré était dû à un problème de friction entre les pièces. Sans rien avoir modifié à notre montage, d'un jour à l'autre un de nos rails ne marchait plus. Bien que le moteur s'active et essaie de tourner, le goal ne se déplaçait pas sur l'axe.

Après avoir inspecté le problème sur plusieurs jours on s'est rendu compte qu'il ne s'agissait ni d'un problème numérique ni électronique. En collant des pièces comme le solénoïde ou le moteur, de la colle s'était infiltrée dans la tige fileté. Pour y remédier, nous avons juste à nettoyer et lubrifier les surfaces.

Bibliographie:

Construction du rail:

Utilisation solénoïdes: <https://www.makerguides.com/control-a-solenoid-with-arduino/>