

Rapport de Projet

BIZOT Patrick
LUCENAY Léonard

Introduction :

Notre projet Arduino avait pour objectif de réaliser une console de jeu jouable par manette sans fil.

Pour obtenir ce résultat, notre binôme s'est fixé divers objectifs annexes. Nous avons partiellement réussi à mener à bien ces objectifs.

On présentera donc dans ce rapport :

- les étapes que nous estimions nécessaires à l'aboutissement du projet
- les schémas électriques de nos montages
- l'algorithme du fonctionnement de la console (dans les grandes lignes)
- l'aspect logistique de notre projet (coût, organisation)
- les problèmes rencontrés et les solutions que nous avons prises ou pourrions prendre (avec plus de temps)

Objectif : Réaliser une console de jeu muni d'une manette sans fil.

Objectifs annexes :

- Communication Bluetooth entre deux Arduino
- Fonctionnement de l'écran
- Manette de jeu (sans fil)
- Support d'écran (boîtier de console)
- Jeux rétros

Coût du projet :

Coût matériel : ≈130€ si on compte la manette VS ≈60€ sans

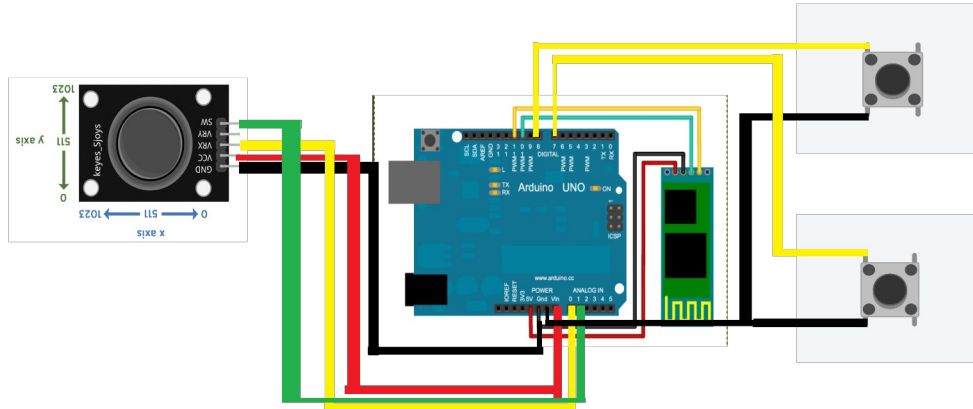
Coût main d'œuvre : ≈30heures(perso) + cours (27h) => ~142€

Total : 272€ avec manette VS 202€ sans

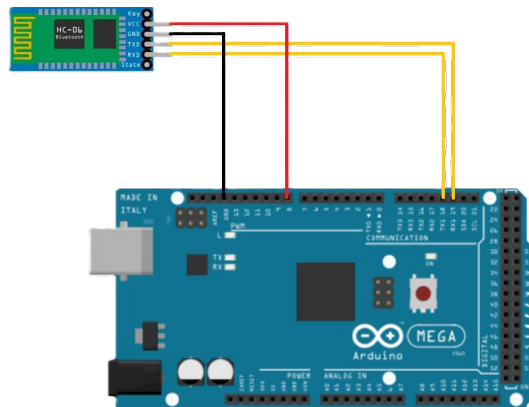
(on a fait une faute en comptant que le coût matériel pendant la présentation)

Schéma Électrique :

Circuit de la manette



Circuit de la console



P.S Le "shield" de l'écran est omis sur le circuit ci-dessus. Il faut tout de même retenir qu'il occupe le 5V de la carte DUE (sur l'image c'est une MEGA) de ce fait l'alimentation du module HC-06 est branchée sur un pin qui sera en HIGH tout le long du fonctionnement (ici c'est le pin 8).

Algorithme de la console :

Les "inputs" du joueur sont entrées à l'aide de la manette puis transmis à la console par Bluetooth. Ensuite, ils sont traités par cette dernière afin de générer une réponse visuelle et fonctionnelle.

Les réponses générées par la console dépendent de l'état de la console :

```
enum class MainState { Menu, Snake, Pong };  
extern MainState main_state;
```

L'état de la console peut lui-même être décomposé en plusieurs sous-états :

```
enum class GameState { Playing, Loading, Paused, Won };  
extern GameState game_state;
```

Selon les états ci-dessus, la boucle principale "switch" d'implémentation.

```
void loop() {  
    switch (main_state) {  
        case MainState::Menu :  
            if (game_state == GameState::Loading) { menu_setup(); }  
            menu_loop();  
            break;  
  
        case MainState::Snake :  
            if (game_state == GameState::Loading) { game_Snake_setup(); }  
            game_Snake_loop();  
            break;  
  
        case MainState::Pong :  
            if (game_state == GameState::Loading) { game_Pong_setup(); }  
            game_Pong_loop();  
            break;  
    }  
}
```

Le code a été fait ainsi afin de faciliter l'ajout de nouveaux jeux.

La seule modification nécessaire est l'ajout d'un nouvel état possible au "MainState" ainsi qu'un cas gérant cet état à la boucle principale.

Par la suite le jeu doit fournir :

- une "loop" pour gérer les inputs de l'utilisateur
- un "setup" pour initialiser l'écran et d'autres variables si besoin

Car les algorithmes de "game_Snake_loop", "game_Pong_loop" et "menu_loop" sont trop grands, nous vous invitons à cliquer [ici](#).

Organisation :

Organisation initiale :

Semaines	S1	S2	S3	S4	S5	S6	S7	S8	S9
Tâches									
Communication BT									
Fonctionnement Ecran									
Ecran d'accueil									
Support Ecran									
Manette Finale									
Jeux									

Organisation finale :

Semaines	S1	S2	S3	S4	S5	S6	S7	S8	S9
Tâches									
Communication BT									
Fonctionnement Ecran									
Ecran d'accueil									
Support Ecran									
Manette Finale									
Jeux									

Légende :

	par Patrick
	par Léonard
	ensemble

Pourquoi un tel écart ?

L'écart avec l'organisation initiale est majoritairement dû au dysfonctionnements du Bluetooth lors des derniers tests, à la re-conception du support de l'écran (car l'ancien ne répondait pas à tout nos besoin) mais aussi dans de moindres mesures à nos difficulté à nous mettre d'accord.

Comment avons-nous surmontés nos problèmes ?

Pour le support de l'écran, nous avons tout simplement repris de zéros et abandonné le design initial bien trop complexe pour ce que l'on souhaite faire.

Pour nos différents durant le projet, cela a été plus compliqué.

Il a fallu faire de concession et garder son sang froid afin d'expliquer au mieux la raison de notre désaccord.

Nos différents reposaient principalement sur le design du support de la console ; l'un était sympa mais ne répondait pas à tout nos besoins l'autre était pas très beau mais répondait à tous.

En ce qui concerne le problème avec le Bluetooth, il s'agissait du fait que les messages envoyés par la manette n'étaient compris que **très rarement** par la console. Nous n'avons pas réussi à surmonter ce problème dans le sens où la manette fonctionne plus actuellement.

Néanmoins, afin de présenter quelque chose de fonctionnel, nous avons décidé de utiliser un seul HC-06 et de l'appairer à notre téléphone.

L'application utilisée était "[Arduino Bluetooth Controller](#)" (les boutons sont à présent **a** ⇒ **carré** et **b** ⇒ **croix**).

Et enfin, cela n'a pas encore été mentionné mais nous avons été incapable de réaliser tous les jeux que nous souhaitions.

La raison principal de cet échec est le manque de temps.

En effet, avec 2 à 3 semaines supplémentaires on aurait facilement pu faire le "Snake" car on avait déjà un prototype non testé d'implémentation.

Conclusion :

Au cours de ces 9 séances de projet, nous avons réalisé en binôme une console de jeu maison contrôlable par téléphone composée :

- d'un support fait par conception 3D
- d'une carte Arduino DUE et UNO avec un écran LCD
- d'une manette reconditionnée
- d'un circuit imprimé à trous fait sur-mesure aller dans la manette

Sur notre console, nous avons réalisé un "Pong".

Avec 9 semaines supplémentaire nous aurions commandé de nouveaux modules Bluetooth (HC-05 et HC-06) pour avoir une communication à nouveau fonctionnelle entre l'écran et la manette de jeu.

Nous aurions réalisé un circuit imprimé et utilisé une NANO (à la place de la UNO) pour occuper moins de place dans la manette et obtenir un résultat plus léger et ergonomique.

Nous aurions fini le code du "Snake" et nous nous serions procuré un écran plus grand et de forme originale afin d'avoir un design unique de console.

Bibliographie

Durant notre projet, nous avons consulté de nombreux sites internet.

Voici quelques liens qui nous avons été amenés à consulter plus d'une fois :

- [Documentation UTFT](#)
- [Site explicatif sur l'appairage HC-05 avec HC-06](#)
- [HC-06 DataSheet](#)
- [Forum Arduino](#)