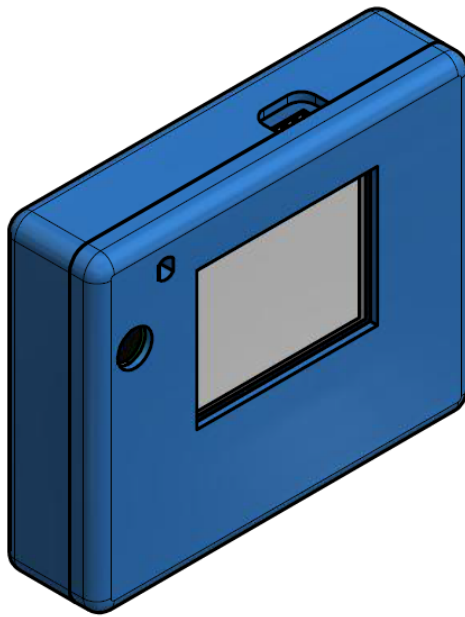


HAUTE ÉCOLE D'INGÉNIEUR DU VALAIS

PONGBOY



SUMMER SCHOOL 2

Table des matières

1	Introduction	2
1.1	Contexte	2
1.2	Spécifications	2
2	Conception	3
2.1	Développement Hardware	3
2.1.1	Alimentation	4
2.1.2	Micro Controlleur	5
2.1.3	Écran résistif tactile LCD	6
2.1.4	Périphériques	7
2.2	Boîtier mécanique	9
2.3	Développement Firmware	10
2.3.1	TouchScreen	11
2.3.2	Mise en veille	12
2.3.3	Contrôleurs	13
2.3.4	Paddle	14
3	Résultats et tests	15
3.1	Résultats	15
3.2	Test	16
4	Conclusion	17
4.1	Bilan	17
4.2	Points à corriger	17
4.3	Voies d'amélioration	17
4.4	Signatures	17
A	Protocoles de test	18

Chapitre 1

Introduction

1.1 Contexte

Ce projet a été réalisé dans le cadre du cours *Summer School 2 - Infotronic* de la Haute école d'ingénieur du Valais. Ce cours de 3 semaines se passe à la fin de l'été, avant la reprise de septembre. Il a pour but de mettre en pratique les compétences acquises durant la 2e année du bachelor. L'objectif du projet est de développer une console de jeu portable, avec écran tactile, permettant de jouer au jeu *Pong*.

Normalement, des équipes de 2 personnes sont formées, mais dû au fait que nous sommes un nombre impair, je me suis retrouvé à développer le PongBoy seul.

1.2 Spécifications

- La console doit être alimentée par 2 piles AAA en série, et ainsi fonctionner en 3[V].
- Une protection contre l'inversion de polarité des piles doit être présente.
- La console doit pouvoir être mise en veille ultra-basse-consommation, pour éviter de changer les piles AAA trop régulièrement.
- L'écran tactile résistif NHD-2.4-240320SF-CTXL-FTN1 doit être utilisé, et une interface tactile doit être implémentée.
- L'utilisation d'une puce driver pour l'écran tactile est prohibée, une librairie driver est fournie.
- Le micro-contrôleur PIC18LF25K22 doit être utilisé, au format SOIC, pour ses capacités d'ultra-basse-consommation.
- L'alimentation de l'écran LCD ainsi que de son rétro-éclairage doit pouvoir être coupée pour respecter la contrainte d'ultra-basse-consommation.

Chapitre 2

Conception

2.1 Développement Hardware

L'architecture du Hardware se présente ainsi :

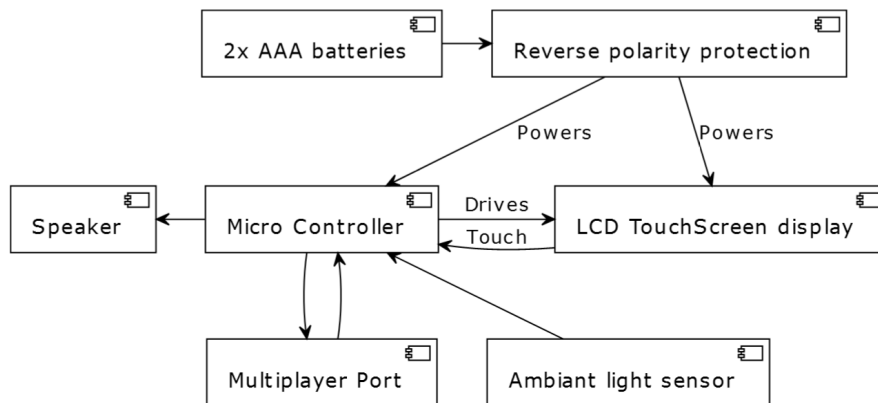


FIGURE 2.1 – Architecture hardware de PongBoy

Le circuit est alimenté par 2 piles AAA, et est protégé contre l'inversion de polarité. Le micro controller commande et récupère des infos de l'écran, de l'haut-parleur, du capteur de luminosité ambiante, et du port multi-joueur.

2.1.1 Alimentation

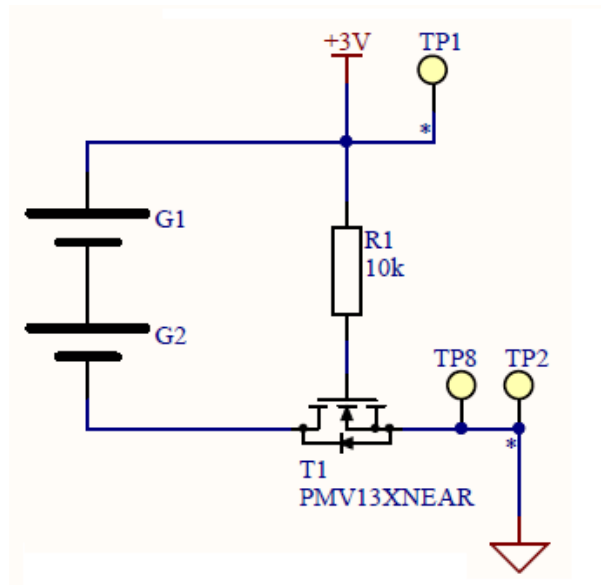


FIGURE 2.2 – Alimentation avec protection contre polarité inverse

La protection d'inversion de polarité est un simple N-MOS qui déconnecte la masse si la tension est inversée. Cette solution est un dérivé de la méthode bien connue du P-MOS sur le VCC. Le P-MOS est souvent plus cher et moins efficace que le N-MOS, ce qui est un problème sur notre circuit basse consommation.

2.1.2 Micro Contrôleur

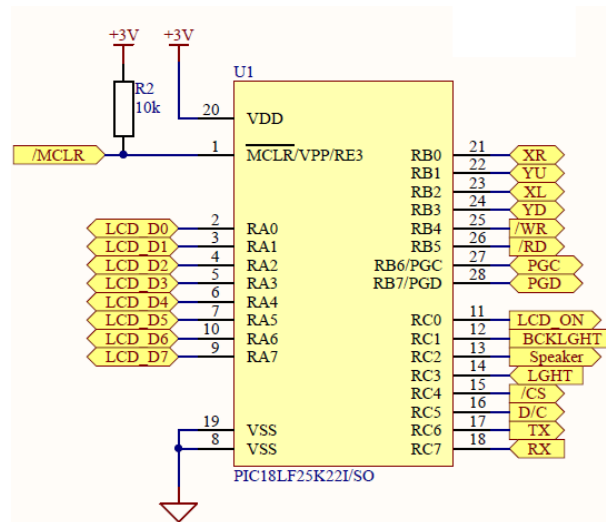


FIGURE 2.3 – Micro contrôleur PIC18LF25K22 avec attribution des GPIO

Le micro contrôleur utilisé est plutôt limité en GPIO, seulement 3 PORT sont disponibles, mais 3 PORT complet. Le bus parallèle de l'écran LCD est entièrement attribué au PORTA. Les 4 résistances de la dalle tactile sont disposée sur le PORTB, sur les pins 0 à 3, choisies pour leur capacités pull-up, d'interruption externe, et de mesure analogique. Le contrôle du rétroéclairage (BCKLGHT) est sur un GPIO avec capacité PWM. PGC, PGD, TX, RX sont préassignée. Le reste n'a pas besoin de capacité spécifique.

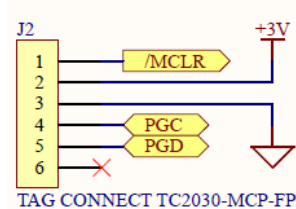


FIGURE 2.4 – Connecteur de programmation du PIC18LF25K22

Le connecteur de programmation est un connecteur TagConnect. Le layout du connecteur est celui du PicKit3.

2.1.3 Écran résistif tactile LCD

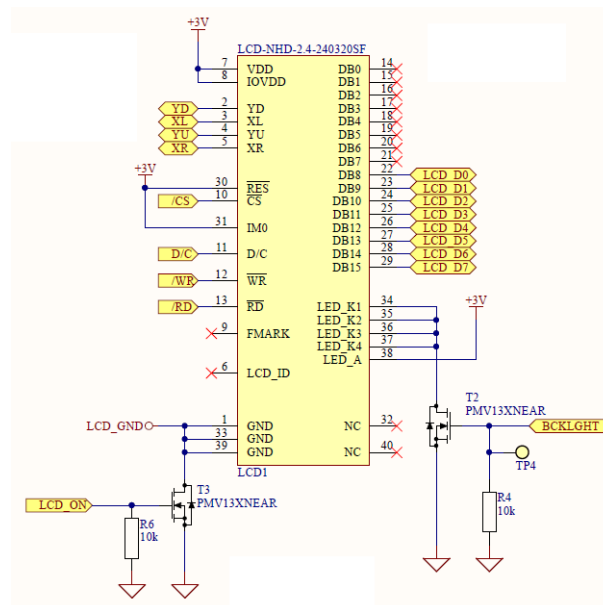


FIGURE 2.5 – Connecteur de l'écran tactile LCD

Des MOSFET sont présent pour couper l'alimentation de l'écran. Étant donné que les MOSFET coupent la masse, il faut s'assurer qu'aucune des pins de l'écran ne soit à la masse lorsqu'on coupe l'alimentation. Si une pin se retrouve à la masse, un phénomène appelé *Puit de courant* se produit, et l'écran arrive à s'alimenter avec cette pin.

Pour ce qui est de la détection et mesure de la pression tactile, le principe suivant sera utilisé sur les pins XR, XL, YU, YD :

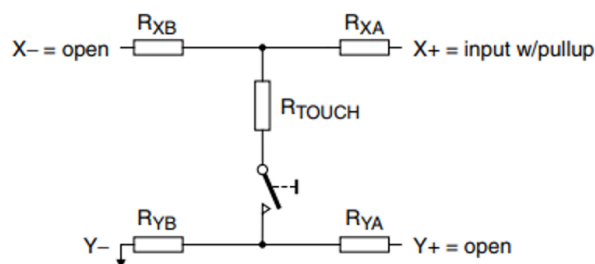


FIGURE 2.6 – Détection de la pression tactile

Pour détecter la pression, on se sert de la dalle tactile comme d'un bouton poussoir, qui va tirer une pin à la masse. Cela requiert donc une pull-up.

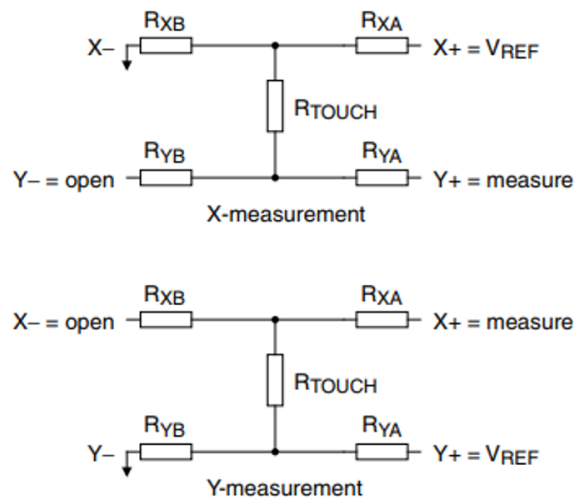


FIGURE 2.7 – Mesure de la position de la pression tactile

Pour mesurer la position de la pression, on se sert du pont présent entre le X et le Y pour faire la mesure. Ainsi, pour mesurer la position en X, nous mesurons une des pins Y, et trouvons donc la tension du pont-diviseur X. Et vice-versa pour la position Y.

2.1.4 Périphériques

Haut-parleur

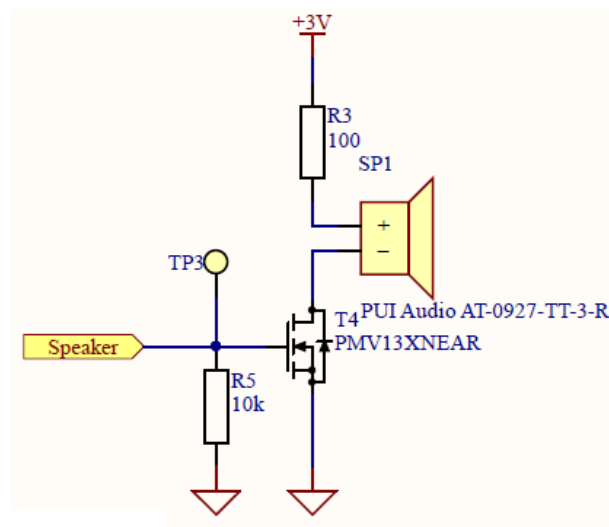


FIGURE 2.8 – Haut-parleur avec MOSFET

Le haut-parleur est piloté avec un MOSFET, pour éviter de faire passer du dans le micro controlleur. Il s'agit d'un haut-parleur inductif, il n'est donc pas nécessaire d'ajouter de diode ou autre composant pour le fonctionnement lorsque le MOSFET coupe le circuit.

Capteur de luminosité ambiante

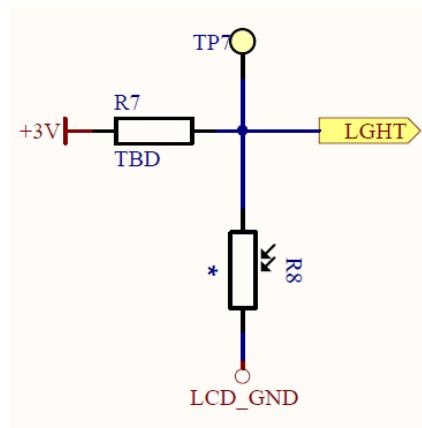


FIGURE 2.9 – Mesure de la lumière ambiante

La mesure de lumière ambiante est réalisée avec une *Light dependant resistor*. La résistance varie en fonction de la lumière, et cette variation est mesurée avec un pont-diviseur de tension. Cette fonctionnalité n'est pour le moment pas du tout implémentée.

Port multi-joueur

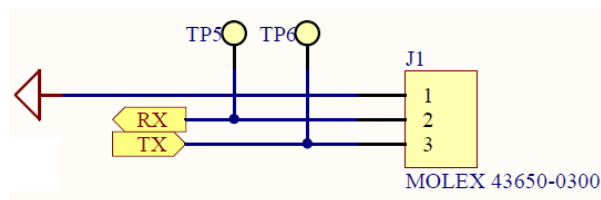


FIGURE 2.10 – Connecteur de communication pour mode multi-joueur

Un connecteur 3-MOLEX a été choisi, pour avoir une communication full-duplex, et transmettre une masse commune. Cette fonctionnalité n'est pour le moment pas du tout implémentée.

2.2 Boîtier mécanique

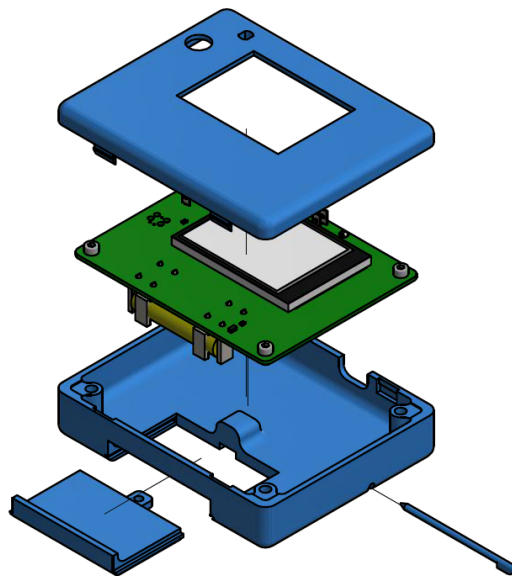


FIGURE 2.11 – Vue éclatée du boîtier complet

Le boîtier est composée de deux parties principales : La plaque inférieur, sur laquelle le circuit imprimé viens se fixer. Et la plaque supérieur, qui se *clips* sur la plaque inférieur. Le circuit imprimé est fixé à l'aide de vis M3, et d'inserts filetés.

Un point d'accès aux piles est présent, pour faciliter le changement de piles. Un stylet a été pensé, pour éviter que l'utilisation du doigt pour le tactile ne bloque la visibilité de l'écran. Il permet également une utilisation plus précise du tactile. Le stylet est inspiré de celui d'une *Nintendo DS*.

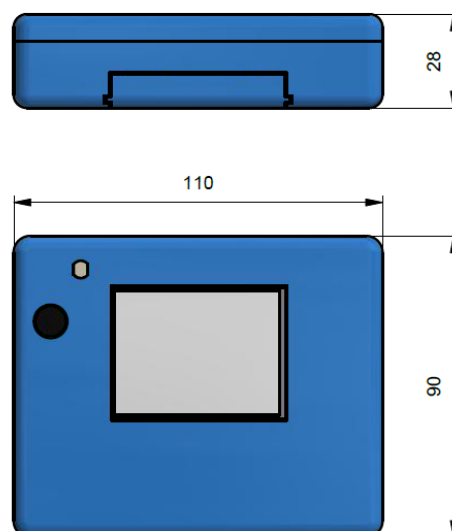


FIGURE 2.12 – Dimensions d'encombrement du bôitier

2.3 Développement Firmware

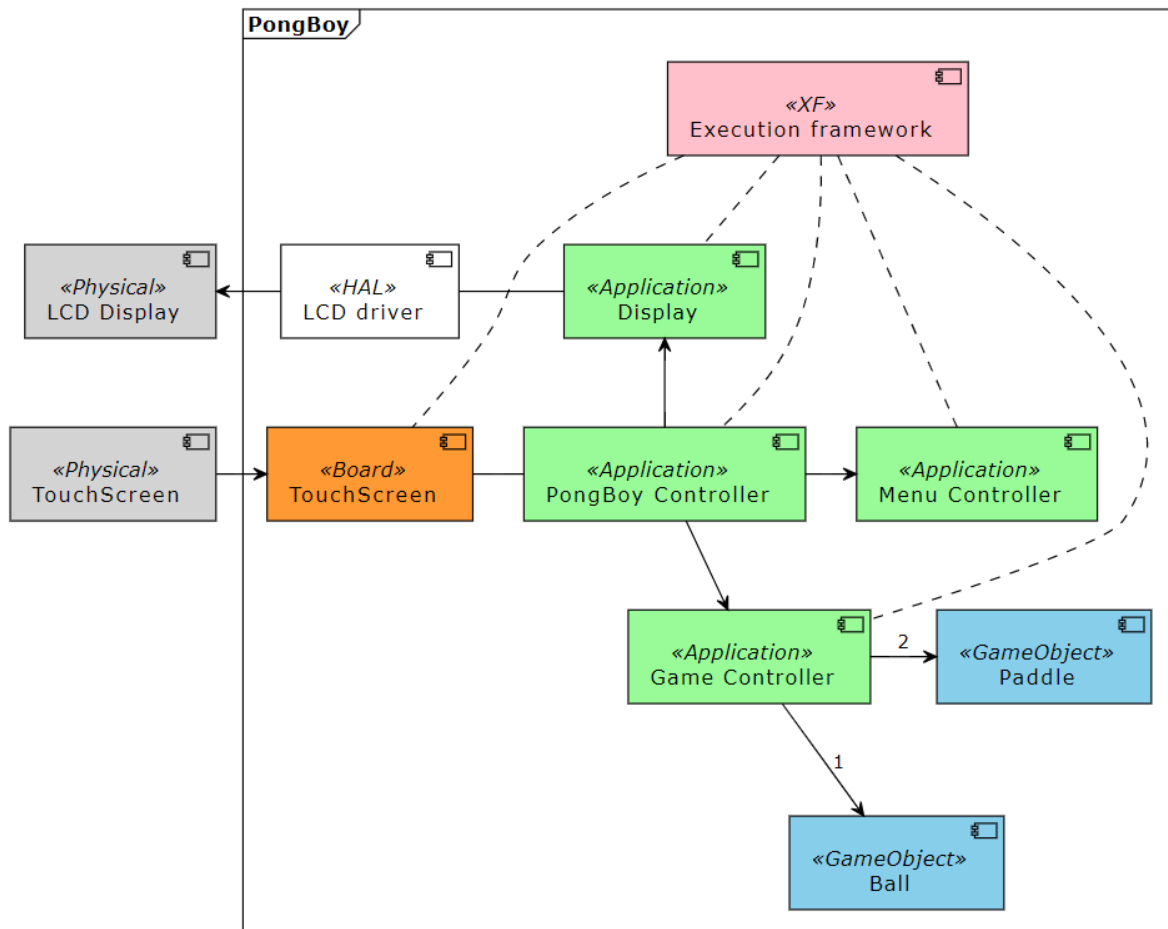


FIGURE 2.13 – Architecture firmware

Le firmware est découpé en plusieurs modules s'occupant chacun d'un élément ou d'une partie du PongBoy spécifique :

- le LCD driver s'occupe d'afficher sur l'écran, et de gérer le rétroéclairage
- TouchScreen s'occupe de détecter une pression sur l'écran, et de mesurer la position de cette pression pour la transmettre plus loin
- Display s'occupe de rafraîchir l'affichage toutes les 30ms, il est informé des changements de menu vers sous-menu, ou vers le jeu
- PongBoy controller est une *Factory*, il contient simplement tous les contrôleurs de la console, comme Display ou Game Controller. Il s'occupe également de la mise en veille de la console.
- Menu Controller et Game Controller font le lien entre les commandes de l'utilisateur, et le menu ou jeu correspondant
- Les *GameObjects* Ball et Paddle sont les balles et raquette du jeu Pong

Le lien entre tous ces modules est le gestionnaire d'événement XF. Chaque module peut envoyer un événement dans la file d'attente de XF, ou programmer un timer qui transmettra un événement. Cette file d'attente est vidée progressivement, et les événements qui en sortent sont retransmis à tous les modules, qui modifieront leur comportement selon l'événement. Un exemple simple : Le jeu se ferme lorsqu'il reçoit l'événement GameOver.

2.3.1 TouchScreen

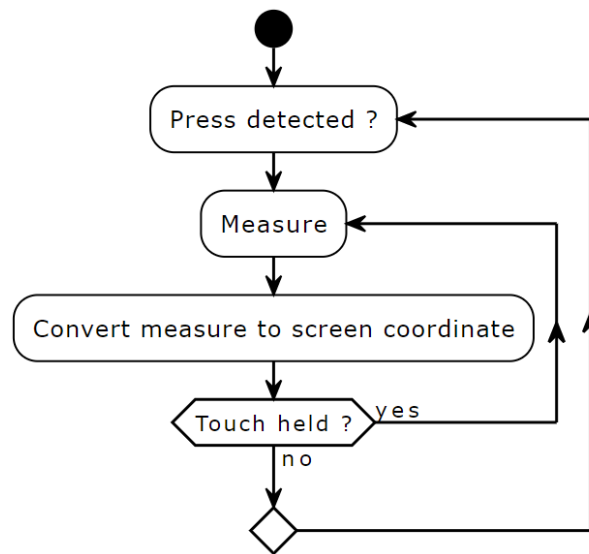


FIGURE 2.14 – Diagramme d’activité du TouchScreen

La détection de la pression et la méthode de mesure est détaillée au point 2.1.3. Le TouchScreen ne mesure que quand une pression est détectée. Les mesures sont espacées entre elles à l’aide d’un timer. À la fin de chaque conversion, le TouchScreen envoie un événement signalant une nouvelle mesure. L’interruption externe du tactile permet de transmettre des événements *Press* et *Release*.

2.3.2 Mise en veille

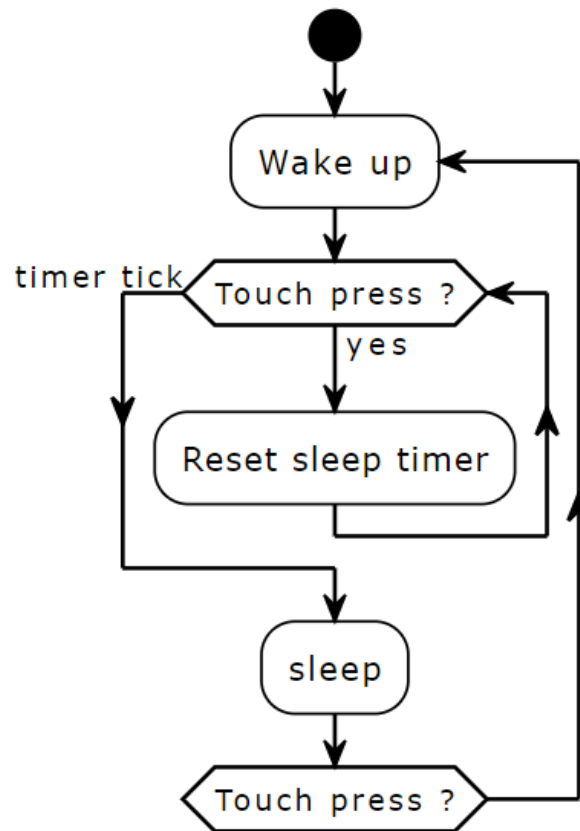


FIGURE 2.15 – Diagramme d'activité du PongBoy - fonction veille

La mise en veille est gérée avec un timer de XF de 20s. À chaque appui sur l'écran tactile, le timer est supprimé et un nouveau est relancé.

2.3.3 Contrôleurs

Pong Controller

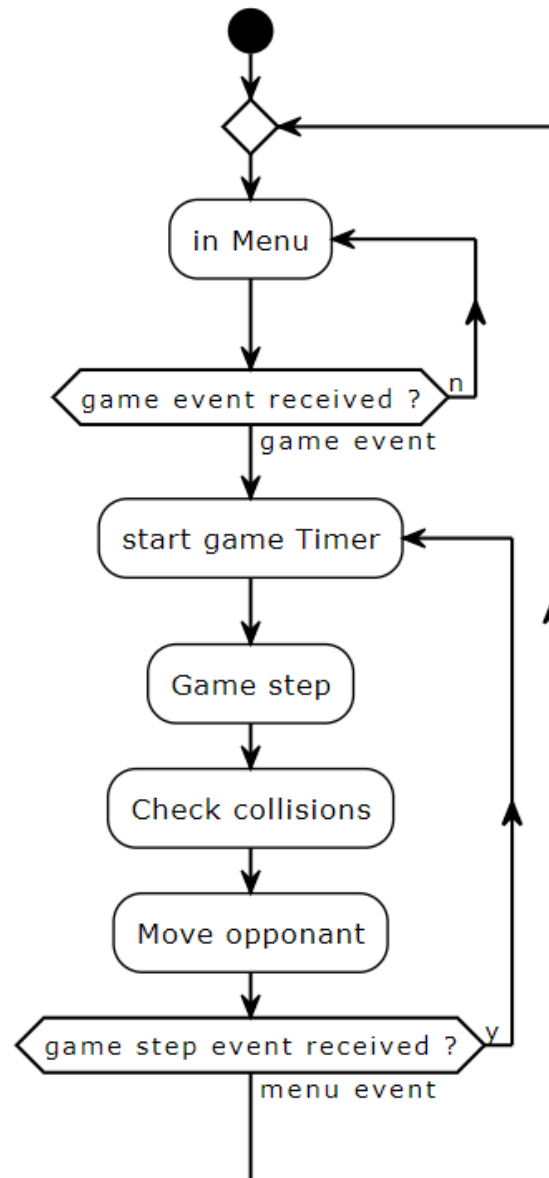


FIGURE 2.16 – Diagramme d'activité du Pong Controller

Le contrôleur de jeu différencie deux états, en jeu et dans le menu. Lorsqu'on est dans le menu, le jeu ne fait rien. Lorsqu'on est en jeu, un timer est lancé pour gérer la physique du jeu. L'affichage du jeu est découplé de ce contrôleur, l'afficheur sait simplement qu'on est en jeu, et il affiche les objets du jeux sur l'écran toutes les $30ms$.

Menu Controller

Le contrôleur de menu fonctionne de manière similaire au pong, sans la partie jeu. Il comprend des états de sous-menu.

2.3.4 Paddle

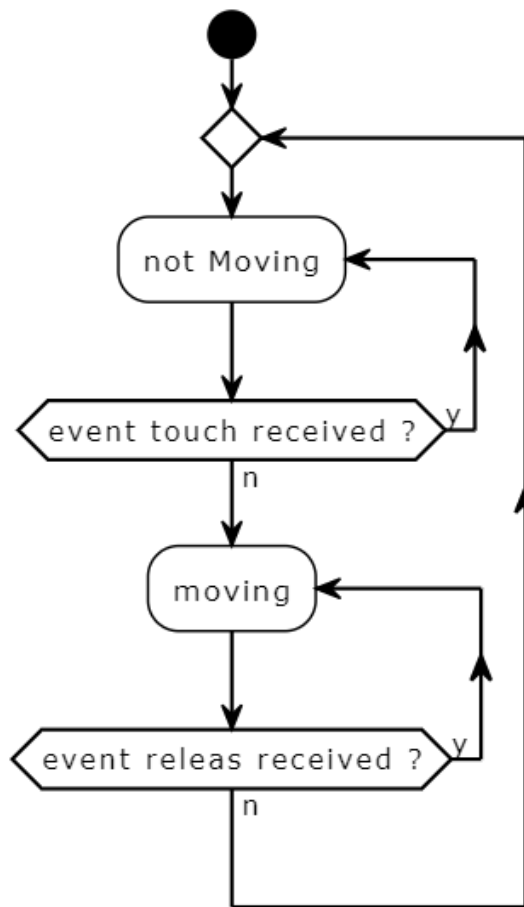


FIGURE 2.17 – Diagramme d'activité du Paddle

La gestion du mouvement du Paddle joueur est découplée du contrôleur de jeu. Le contrôleur de jeu transmet les événements au Paddle. Si une pression sur l'écran a été détectée, le Paddle va checker si la pression est sur la partie du haut ou du bas de l'écran, et va modifier sa vitesse en adéquation.

Chapitre 3

Résultats et tests

3.1 Résultats

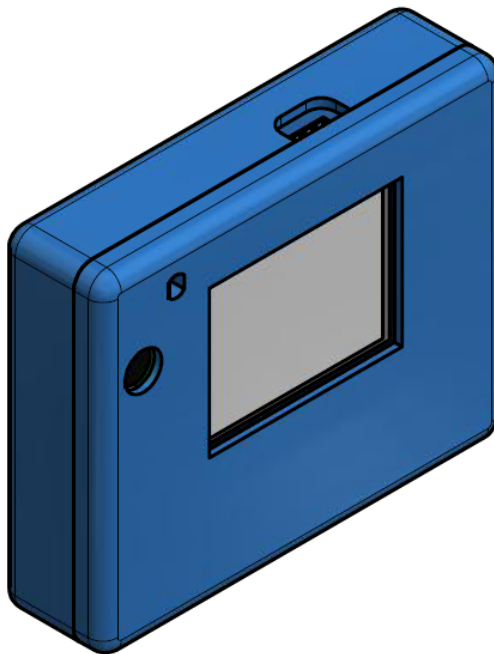


FIGURE 3.1 – Boîtier complet du PongBoy

Le cahier des charges est respecté. La jouabilité est correcte. Par manque de temps, et manque de ressources humaines, les fonctionnalités comme le haut-parleur, et le mode multi-joueur n'ont pas pu être implémentés.

3.2 Test

De nombreux tests ont été réalisés pour vérifier la qualité du produit. Ces tests ont permis de détecter de nombreux problèmes, qui ont bien entendu été corrigés.

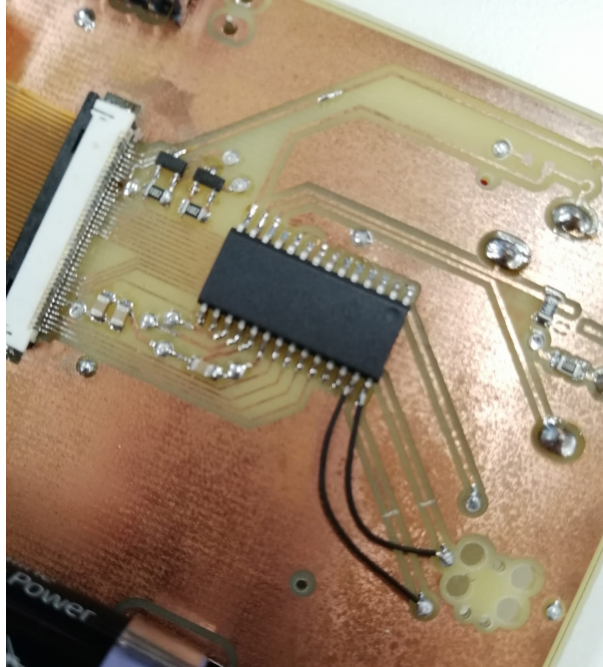


FIGURE 3.2 – Signaux de programmation PGC et PGD inversés

Par exemple, les signaux PGC et PGD du port de programmation ont été inversé dans la schématique, résultant en une impossibilité de programmer le processeur. Cela a été corrigé comme montré dans la figure 3.2

L'ensemble des tests réalisés durant ce projet sont disponibles en Annexe A.

Chapitre 4

Conclusion

4.1 Bilan

Le cahier des charges a été respecté dans son intégralité. Le mode veille fonctionne consomme $9[\mu A]$. Le gameplay est proche du vrai Pong, l'utilisation du stylet rend le gameplay plus agréable qu'au doigt. Le réglage du rétroéclairage fonctionne parfaitement, et permet aussi meilleure autonomie.

4.2 Points à corriger

- Sur le PCB v1.0, les signaux PGC et PGD ont été inversés sur le port de programmation.
- Sur le PCB v1.0 Les connecteurs de piles sont trop proches l'un de l'autre, du scotch. isolant a dû être placé pour éviter des courts-circuits.
- Sur le boîtier v2.1, les fixations de la plaque supérieur sont trop fragiles.
- Sur le boîtier v2.1, l'accès aux piles est trop étroit et rend le changement de pile avec les doigts très désagréable.
- Sur le boîtier v2.1, le stylet ne peut pas rentrer correctement dans le boîtier à cause de l'impression 3D imprécise.
- Les pull-downs présentent sur les Gate des MOSFET ne sont pas nécessaires. Le PIC18LF25K22, lorsqu'il part en veille, conserve l'état de ses entrées/sorties d'avant veille.

4.3 Voies d'amélioration

- Ajout d'un joystick ou slider pour le contrôle du Paddle.
- Amélioration du menu du PongBoy, ajout d'informations visuelles pour l'utilisateur (par ex : le slider de luminosité n'a pas de légende).
- Ajout de plusieurs réglages dans le menu paramètres (par ex : choisir de jouer sur fond blanc plutôt que fond noir).
- Amélioration du gameplay (balle qui s'accélère durant la manche).
- Ajout d'un choix de difficulté de l'adversaire dans Pong.
- Implémentation du haut-parleur, ajout de feedback sonore dans le menu et dans le Pong.
- Implémentation du capteur de luminosité ambiante, et réglage automatique du rétroéclairage.
- Implémentation du mode multi-joueur.

4.4 Signatures

Samy Francelet

Annexe A

Protocoles de test

Test ID	Hardware - 01
Description	Inspection des défauts de fabrication
Pattern	White Box
Test Description	Le PCB, juste après la fabrication et l'usinage, est inspecté à la loupe dans le but de rechercher d'éventuelle court circuit entre les pistes ou alors des pistes coupées qui aurait été rongée par l'acide
Personnel	Samy Francelet
Results	2 pistes coupées : PGC et PGD
Conclusions	Le cuivre de la face supérieur a été plus mangé par l'acide, les pistes fines du port de programmation en ont souffert
Measures	Un morceau de fil conducteur a été soudé par-dessus la piste

Test ID	Hardware - 02
Description	Ce test permet de vérifier si le PCB n'a pas de court-circuit
Pattern	Smoke test
Test Description	Le PCB assemblé complet va être inspecté à la loupe, puis alimenté en 3V, avec une limitation de courant <150mA pour éviter de faire de trop gros dégats en cas de court-circuit.
Personnel	Samy Francelet
Results	Pas de court-circuit observé. Le courant est <10mA.
Conclusions	Ce courant correspond à la consommation du PIC n'étant pas encore programmé.
Measures	Le test est réussi.

Test ID	Hardware - 03
Description	Ce test permet de vérifier la protection contre l'inversion de polarité
Pattern	Smoke test
Test Description	On inverse la polarité de l'alimentation afin de voir si notre protection fonctionne correctement
Personnel	Samy Francelet
Results	Pas d'échauffement de composant. Pas de courant
Conclusions	La protection d'alimentation fonctionne correctement
Measures	Le test est réussi.

Test ID	Firmware - 01
Description	Programmabilité du micro contrôleur
Pattern	White Box
Test Description	On lance MPLAB puis on crée un nouveau projet avec le bon compilateur (XC8) et le bon PIC (18LF25k22). Le PCB vient se connecter avec le connecteur de programmation sur l'ordinateur. Une première compilation et programmation est lancée. Si le programme s'arrête avec une erreur, on débranche le Pickit3 et on cherche la faute. Sinon, le test prend fin.
Personnel	Samy Francelet
Results	MPLAB ne reconnait pas le micro contrôleur
Conclusions	Les pins PGC et PGD sont inversées
Measures	Les deux pistes sont coupées et deux fils sont utilisés à la place

Test ID	Firmware - 02
Description	Changement de couleur de fond d'écran
Pattern	White Box
Test Description	Vérifie que la commande tactile fonctionne, et que la communication avec l'écran fonctionne
Personnel	Samy Francelet
Results	Fonctionnel
Conclusions	Le test est réussi
Measures	

Test ID	Firmware - 03
Description	Balle qui rebondit contre les parois de l'écran
Pattern	White Box
Test Description	Vérifie que le contrôleur de jeu fonctionne et que les collisions sont correctes
Personnel	Samy Francelet
Results	Aucun bug de rebond
Conclusions	Le test est réussi
Measures	

Test ID	Firmware - 04
Description	Test du sleep
Pattern	White Box
turn	On vérifie que l'appareil s'éteint bien et bien après 20 sec
Personnel	Samy Francelet
Results	L'appareil s'éteint correctement
Conclusions	Le test est réussi
Measures	

Test ID	Firmware - 05
Description	Stress test du jeu Pong
Pattern	Black box
Test Description	Vérifier que le programme répond correctement, même si plusieurs utilisateurs viennent essayer le jeu, faire des commandes absurdes, appuyer avec 12 doigts, etc..
Personnel	Samy Francelet
Results	Aucun bug détecté
Conclusions	Le test est réussi
Measures	