


# Pad d'arcade

Une manette type arcade utilisant le protocole USBHID.

 Difficulté **Difficile**

 Durée **1 jour(s)**

 Catégories **Jeux & Loisirs**

 Coût **50 EUR (€)**

## Sommaire

Introduction

Étape 1 - Circuit imprimé

Étape 2 - Programmation du PIC

Étape 3 - Boîtier

Étape 4 - Résultat final

Notes et références

Commentaires

## Introduction

Le projet fini, dans la version 1.1, propose un Pad arcade compatible Windows (testé sous xp, et 7), non testé sous Mac,Linux. Le protocole USB utilisé est le "HID", ne nécessitant pas de pilote. Le circuit peut supporter soit :

- 2 axes numériques + 24 boutons (2 joueurs arcade)

- 2 axes analogiques + 10 boutons (manette commerciale la plus fréquente)

Le support (boîtier) quand à lui est conçu pour porter une Pad arcade 1 joueur.

## Matériaux

- Circuit imprimé (cf. partie circuit)
- Pic 18f4550
- Résistances (3,3k<sub>(2)</sub> et 3,3k -> 10k ohm<sub>(14 pour le pad 1 joueur)</sub>)
- Capacité (condensateurs 47nf et 100uF)
- Bois (MDF 5 (ou 6mm))
- Câble USB<sub>A</sub> mâle
- Bouton et stick
- Pickit 3 (ou autre programmeur)


## Outils

### Machines utilisées

- Machine d'impression de circuit (CNC ou imprimante chimique).
- Découpeuse laser
- Fer à souder

### Logiciel utilisé

- MPLab IDE 8.91 disponible sur le site de microchip
- Eagle ou autre logiciel de CAO

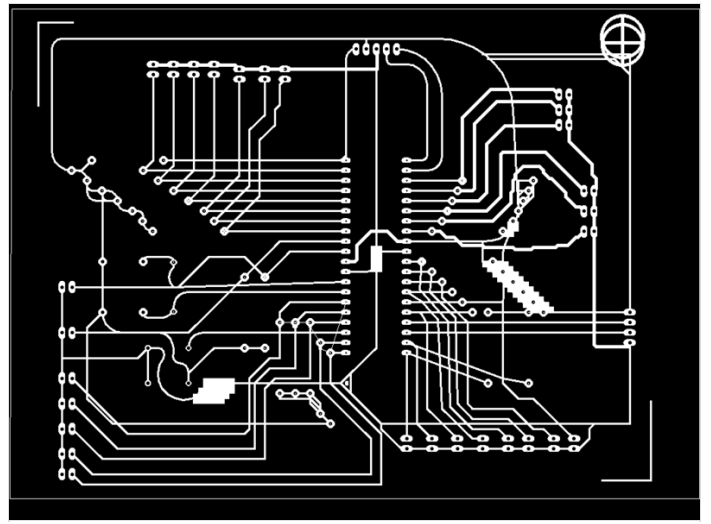
 [https://github.com/hugotpichon/arcpad-pic18f4550/tree/master/3D\\_2D/fichier%202D%20vect](https://github.com/hugotpichon/arcpad-pic18f4550/tree/master/3D_2D/fichier%202D%20vect)

# Étape 1 - Circuit imprimé

Pour ce qui est du circuit imprimé, je vous propose celui que j'ai conçu.

Il est conseillé d'apporter les améliorations suivantes :

- Ajout d'un plan de masse (pour une gravure du circuit à la CNC, et pour faire moins de bruit CEM)
- Réduction de la taille (coût plus faible et moins de place dans la boîte)



# Étape 2 - Programmation du PIC

Pour le Pic18f4550 un programmeur Pickit 3 (~15€) suffit.

Deux logiciels s'offrent à vous :

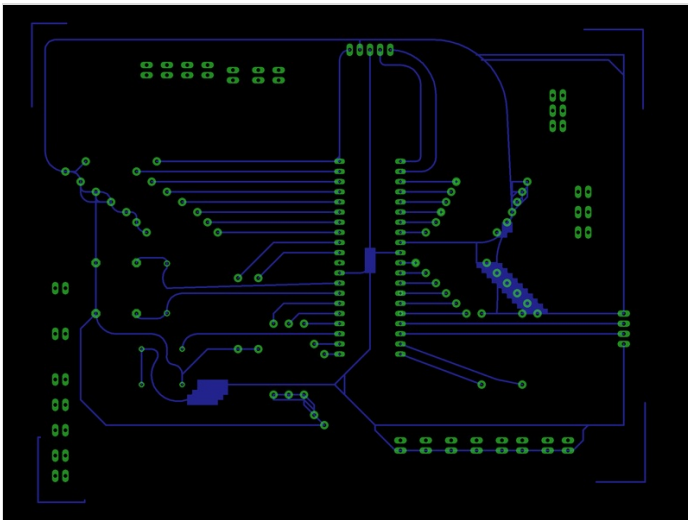
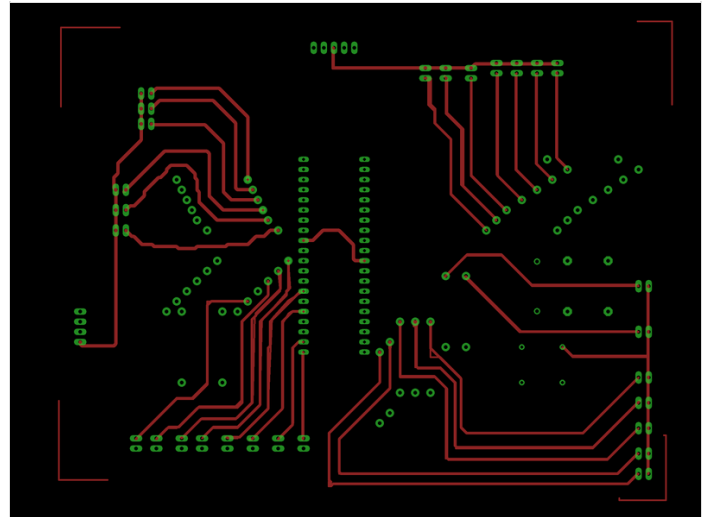
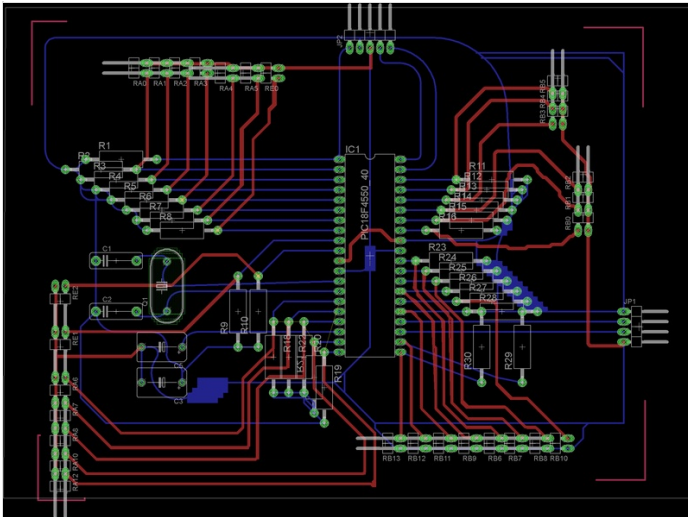
- MPLab 8.92 + c18 compiler
- MPLab X + xc8 compiler

Pour une simple raison de programme d'exemple disponible et de prise en main j'ai utilisé MPLab 8.91 + c18 compiler (disponible sur le site de Microchip)

J'ai donc modifié le fichier d'exemple "HID - Mousse" pour en faire un HID - joystick

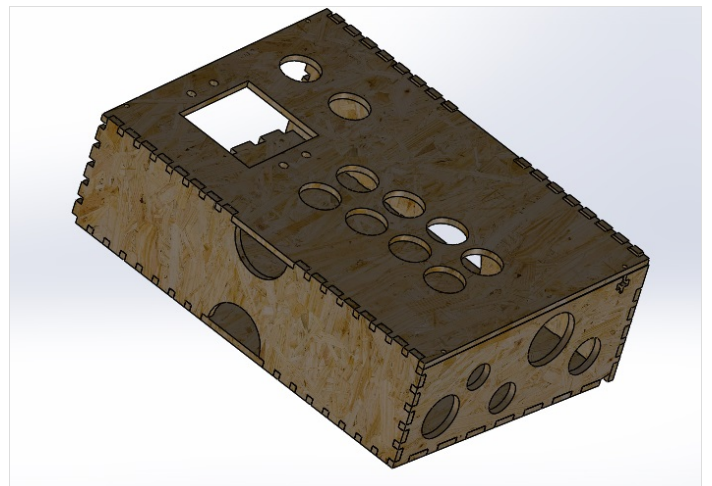
Les fichiers qui vous seront utiles pour modification sont :

- joypad.c
  - #pragma config PLLDIV = X //(X = fréquence du quartz/4Mhz)
  - char buffer[7]; // peut être modifié si vous changez les sorties de la manette
  - void joypad(void) //à modifier si vous changez les sorties
- usb\_config.h
  - #define HID\_INT\_IN\_EP\_SIZE 7 //nombre d'octets par message
  - #define HID\_RPT01\_SIZE 55 // à changer si vous modifiez le "HID report"
  - #define MY\_VID 0x04D8 #define MY\_PID 0x80FA // éditez si vous avez mieux, ce sont des VID et PID de microchip (non commerciaux)
- joypad\_map.c
  - char Lect\_ANX(void)
    - N'est pas utilisé, mais quelques changements permettent d'effectuer la lecture d'un stick analogique.
  - char Lect\_XY(void)
    - Permet la lecture de stick numérique (type arcade)
  - char Lect\_bouton\_X\_to\_Y (void)
    - Permet la lecture des boutons
    - Vous devrez probablement modifier les valeurs : if (boutonX == 0) en fonction de vos soudures.
- joypad\_map.h
  - Définit les utilitaires liés à joypad\_map.c
    - Pratique pour repérer les liens entre "entrée et bouton"
- usb\_descriptor.c
  - Le fichier le plus complexe que vous pourriez avoir à modifier
  - //Manufacturer string descriptor
    - votre nom, plus ou moins
      - n'oubliez pas de modifier "string[25]" si vous changez le nombre de caractère
  - //Product string descriptor
    - Le nom qu'il aura dans le système (vue par les jeux et autres)
      - N'oubliez pas de modifier "string[25]" si vous changez le nombre de caractères
        - (un nombre réduit de caractères < 12, sans accents, est une bonne idée)
  - //Class specific descriptor - HID joystick
    - Bien que je vous déconseille de vous aventurer en terrain dangereux vous devrez toucher à cette horreur si vous souhaitez modifier les sorties du pad.
    - Je n'ai pas tout compris à ce langage mais si vous souhaitez vous y aventurer :
      - "usb\_lyser" (payant mais version trial de 30 jours disponible)
        - Peut vous aider en récupérant les HID report des objets que vous connectez en USB à votre PC (souris pas chère, manette,...).
      - MSDEV : Dt.exe (logiciel fourni par Microsoft) permet d'écrire les Bytes(octet) de code du HID report.



## Étape 3 - Boîtier

La boîte a été faite pour 1 joueur.  
Elle contient 10 boutons (8 de jeu + select, start), 1 espace pour le stick et 2 visser pour fermer le capot supérieur. Le capot inférieur est fixé par de la colle à bois aux quatre contours.



## Étape 4 - Résultat final

Bravo votre borne est finie !



---

## Notes et références

Un projet réalisé au fablab LabBoîte de Cergy à retrouver aussi sur [github.com/hugotpichon/arcpad-pic18f4550](https://github.com/hugotpichon/arcpad-pic18f4550).