



CPNV

# Evolution du code manette arcade

P1704\_Manettes

PICOTTE Alexandre  
SANDOZ Pierre-Olivier  
27/04/2018

## Introduction :

La manette arcade possède deux codes : un pour l'arduino leonardo, qui agit en tant que master et l'autre pour les arduinos unos, qui ont le rôle de slave. Ces deux codes ont connu une évolution selon nos besoins et nécessités.

## Première version :

La première version prévoyait d'accueillir jusqu'à quatre joueurs simultanément et mettait les joueurs supplémentaires en attente. Chaque uno lors de sa mise en service envoyait une requête au leonardo pour que ce dernier leur donne une adresse, si le nombre de joueur avait atteint 4 alors, le leonardo envoyait une requête à chaque uno enregistré et dans le cas où un des uno ne répondait pas, le leonardo l'enlevait de sa liste de joueur et prévenait le uno mis en attente qu'une place s'est libérée. Concernant la communication de l'état des boutons, l'arduino uno enregistre les changements d'état, lorsqu'un uno détecte un changement il communique l'état de tous les boutons au leonardo qui enregistre ces changements pour appliquer les effets désirés, relâcher une touche si un bouton n'est plus appuyé et vice versa. Cette trame n'est envoyée qu'une fois par changement détecté par l'arduino uno, c'est aussi l'arduino uno qui communique l'information à l'arduino leonardo, sans que celui-ci ait à le demander.

La trame d'information est composée de deux bytes pour le header, qui permettent d'autoriser la réception de données, puis d'un byte qui contient l'adresse du joueur et enfin 2 bytes qui contiennent les informations concernant l'état des boutons. Pour les boutons, chaque bit est assigné à un bouton, si le bouton est pressé alors le bit passe à 1 et dans le cas contraire il est à 0.

```
//touche clavier = 21 + numéro joueur * 12 + le numéro du bit du premier byte  
//touche clavier = 21 + numéro joueur * 12 + le numéro du bit du deuxième byte + 4
```

Le premier exemple montre le calcul pour le premier byte de l'état des boutons et le deuxième exemple pour le second byte. Selon les références [arduino\(https://www.arduino.cc/en/Reference/ASCIIchart\)](https://www.arduino.cc/en/Reference/ASCIIchart) les 31 premiers caractères sont des caractères de contrôle et le 32<sup>ème</sup> correspond à la barre espace, ainsi on a décidé que le premier bouton du premier joueur sera attribué au 33<sup>ème</sup> caractère. La formule pour calculer la valeur réservée à chaque joueur est une plage de 12 caractères.

Plusieurs problèmes étaient survenus avec ce code. Le premier problème était la « perte » de données, avec seulement un échange de données par changement détecté il arrivait que le leonardo ne réagisse pas. Le deuxième problème vient de la bibliothèque keyboard qui permet d'émuler un clavier. Le code du leonardo attribuait une valeur à chaque touche de chaque joueur, puis grâce à la table ascii convertissait cette valeur en un caractère. Mais les valeurs entrées et les caractères sortis ne correspondaient pas. Lorsque le leonardo écrit les caractères nécessitant la touche shift, il presse d'abord shift puis l'autre touche, ainsi toutes les majuscules et caractères demandant shift sont inutilisables car lors de la configuration de la manette sur l'émulateur, ce dernier n'enregistrera que shift.

## Seconde version :

Nous avons donc dû reprendre le code pour l'améliorer. La deuxième version du code réduisait le nombre de joueur à 2, quatre joueurs posaient problèmes avec l'attribution des touches. On a aussi changé la manière dont le leonardo sélectionnait quelles touches attribuées à quels boutons.

Avant on calculait une valeur qui ensuite était converti selon la table ascii en caractère, maintenant on calcule toujours une valeur mais celle-ci est envoyé dans un switch...case, qui selon la valeur va attribuer directement le caractère.

```
switch (number) {  
  case 33:  
    if (press_release) {  
      Keyboard.press('q');  
    } else {  
      Keyboard.release('q');  
    }  
    break;
```

Cette version nous évite le problème de la conversion par ascii et le problème des caractères qui demande un shift.

Nous avons aussi modifié la fréquence avec laquelle les arduinos communiquent. Avant ils ne communiquaient que lorsqu'un changement était détecté, maintenant le uno envoie en permanence l'état des boutons.

Malheureusement cette version présentait aussi les mêmes soucis que la version précédente, c'est-à-dire que le leonardo ne reçoit pas toujours les informations transmises par les unos, on a aussi noté une réaction assez lente, une latence perceptible par l'être humain, du système.

## Troisième version :

Pour cette version nous avons laissé les changements de la deuxième version sur le nombre de joueur et sur l'attribution des touches. Cependant nous entièrement changé le mode de communication.

Auparavant, les unos communiquaient que les changements, ensuite ils communiquaient l'état des boutons en permanence, finalement on a opté pour que ce soit le leonardo qui réclame les informations à chaque fois qu'il boucle. On a changé la méthode des unos pour les requêtes et changé la méthode du leonardo qui recevait des informations.

Cette version est plus rapide que la seconder version, même si nous n'avons pas fait de mesure la réaction du système était proche d'un clavier standart.