

POLYTECH LILLE

Rapport de projet de fin d'étude

**Hack-a-Wii : Emulation de WiiMote pour
rendre la Wii accessibles aux personnes
handicapées**

Fabien Violier

25/02/2015

Introduction

Ce rapport a pour but de dresser un bilan de mon projet de fin d'étude, effectué en partenariat avec l'IRCICA. Ce projet intervient dans le cadre d'une coopération avec un centre pour handicapé.

Compte tenu de la fenêtre de temps relativement courte dans laquelle s'est effectué le projet, ce rapport reprend majoritairement le contenu de la page Wiki du projet.

Contexte : Les éducateurs utilisent la Wii dans leurs animations. Cependant, certains handicapés ne peuvent tenir la WiiMote (manette de jeu de la Nintendo Wii) correctement. Les éducateurs sont donc obligés de les accompagner dans les gestes, ce qui fausse le jeu ainsi que l'intérêt ludique de l'activité.

But : Afin de pallier à ces problèmes de motricité, plusieurs solutions sont possibles, comme des capteurs sur les vêtements ou l'utilisation d'une kinnect. Pour implémenter ces solutions il faut d'abord pouvoir dialoguer avec la console. C'est dans cet objectif qu'intervient mon projet de fin d'étude qui consiste à émuler la manette de jeu.

Pour émuler cette manette nous avons d'abord approché une solution par la création d'une manette, notamment par l'utilisation de module Bluetooth. Nous avons ensuite approché une méthode par contrôle d'une WiiMote.

Table des matières

Introduction.....	2
Partie 1 - Emulation d'une WiiMote via des modules Bluetooth.....	4
Prise en main du Bluetooth, découverte des modules Bluetooth	4
Les modules Bluetooth.....	4
Le Module série simple d'utilisation.....	4
Le module HC 06.....	4
Le module HC 05.....	5
Configuration de la WiiMote.....	5
Configuration des modules	6
Conclusion	6
Partie 2 : Asservissement d'une WiiMote	7
La manette.....	7
Étape 1 le contrôle des boutons.....	7
Étape 2 l'accéléromètre.....	8
Étape 3 émulation du nunchuck.....	8
Le programme Arduino	8
Le système final.....	10
Les tests.....	10
Conclusion	12

Partie 1 - Emulation d'une WiiMote via des modules Bluetooth

Prise en main du Bluetooth, découverte des modules Bluetooth

Lors de la première réunion du projet, nous avons décidé d'explorer les différents modules Bluetooth usuellement piloté par une plateforme Arduino. Nous avons choisi cette piste car l'Arduino nous permet un contrôle plus "bas niveau" et permettrait un panel de configuration du Bluetooth plus large. L'idée étant de reproduire une WiiMote et d'envoyer les mêmes messages que celle-ci.

Afin que l'illusion soit complète et qu'il n'y ait pas de problème de connexion entre cette reproduction de WiiMote et la Wii, il faut avoir accès à un large panel de configuration comme :

- le nom
- l'adresse MAC
- le code PIN
- la "Class Of Device" (COD)
- ...

Au vu de l'état de l'art actuel dans le domaine, les éléments de configuration indispensables à la connexion n'ont pas pu être listés.

Les modules Bluetooth

Le Module série simple d'utilisation

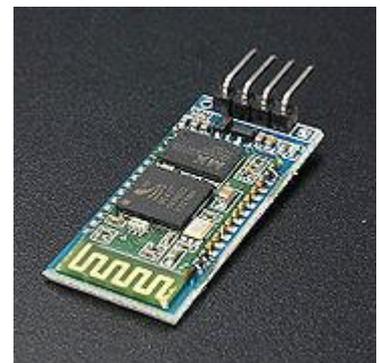
Ce module permet d'établir une liaison série facilement. Il n'y a pas de configuration possible.

Ce module ne répond donc pas à nos besoins. Nous l'avons rapidement abandonnée pour un module offrant davantage de configuration.

Le module HC 06

Ce module permet d'établir une communication série simple, la configuration est rapide et simple. Il n'est cependant pas possible de configurer entièrement le module.

Pour configurer le module, on utilise les commandes AT. Attention toutefois à la tension d'entrée sur la broche Rx, le module accepte du 3.3V il est donc recommandé de faire un pont diviseur de tension sur cette broche.



Le protocole de configuration est le suivant : (pour ce module il ne faut pas envoyer de retour charriot « \r\n », entre chaque commande il est obligatoire d'attendre 2 secondes.

Emission : AT
Attente
Réception : OK

Il est possible de changer le nom du périphérique, le code PIN ainsi que la vitesse de communication, par défaut la vitesse est de 9600BAUD, cette vitesse resterait utilisable peu importe la vitesse choisie.

Changer le nom : AT+NAMEnomduperipherique
répond OKsetname

Changer le code pin : AT+PINxxxx
répond OKsetpin

Changer la vitesse de transmission : AT+BAUDx 'x' allant de 1 à 9

1 correspond à 1200BAUD

4 correspond à 9600 BAUD

Ce module n'est donc pas adapté à l'émulation d'une WiiMote celui-ci n'offrant pas suffisamment de personnalisation du module.

Le module HC 05

Ce module permet d'établir une communication Bluetooth. Ce module permet aussi un large panel de configuration possible. On configure ce module en utilisant des commandes AT.

Le protocole est le suivant : On envoie une commande AT\r\n à une vitesse de 38400 BAUD, La pin « KEY » du module doit être mise à VCC. Le module accepte du 0-5V.



Suite à l'envoi de la commande AT\r\n le module répond OK.

Configuration de la WiiMote

Après avoir connecté la WiiMote à un ordinateur sous Linux, l'appel de la fonction hidtools info nous renvoie les informations suivantes.

```
BD Address: 00:24:1E:B8:87:6F
Device Name: Nintendo RVL-CNT-01
LMP Version: 2.0 (0x3) LMP Subversion: 0x31c
Manufacturer: Broadcom Corporation (15)
Features: 0xbc 0x02 0x04 0x38 0x08 0x00 0x00 0x00
          <encryption> <slot offset> <timing accuracy> <role switch>
          <sniff mode> <RSSI> <power control> <enhanced iscan>
          <interlaced iscan> <interlaced pscan> <AFH cap. slave>
```

Le site internet wiibrew.org/wiki nous apporte des informations complémentaires sur la configuration

```
Wii Remote/old Wii Remote Plus
Name Nintendo RVL-CNT-01
Vendor ID 0x057e
Product ID 0x0306
Major Device Class 1280
Minor Device Class 4
Service Class 0
(Summary of all Class Values) 0x002504
```

Configuration des modules

J'ai essayé de configurer les modules de façon à imiter une WiiMote. Voici la configuration que j'ai utilisée pour le module HC 05.

```
AT+NAME=Nintendo RVL-CNT-01\r\n nom du module
AT+BIND=0024,1E,B8876F\r\n adresse mac du module
AT+CLASS=2504\r\n
Apparaît en manette de jeu lors de la connexion au pc
AT+ROLE=0\r\n (par défaut) slave
AT+PSWD=1234\r\n code PIN du module
AT+CMODE=1\r\n (défaut) le périphérique se connecte sur n'importe quelle
adresse
```

Cette configuration n'a pas permis d'établir une connexion entre la Wii et notre système. Le site wiibrew.org/wiki indique que le mot de passe WiiMote suit l'adresse mac en ASCII. La configuration suivante a également échoué.

```
AT+BIND=3635,34,333231\r\n (ce qui correspond en code ASCII à 65,4,321)
AT+PSWD=123456\r\n
```

Conclusion

La WiiMote étant un périphérique utilisant un protocole HID, il n'est pas possible, ou pas encore réalisé, d'établir une connexion en utilisant les modules précédents, il est nécessaire d'utiliser des modules intégrant de base le protocole. Cependant, à la vue du temps restant allouer au projet, il a été proposé de changer de point de vue et d'essayer de contrôler la WiiMote au lieu de l'émuler.

Partie 2 : Asservissement d'une WiiMote

Afin de contourner le problème rencontré avec les modules Bluetooth je vais maintenant essayer d'asservir une WiiMote. L'idée est donc de brancher un Arduino à la WiiMote et exécuter la commande reçue. Dans un premier temps on ouvrira la WiiMote afin de mettre en évidence les éléments qui nous intéressent. La première étape sera d'activer des boutons. La 2ème d'envoyer des instructions de l'accéléromètre de la WiiMote. La 3ème étape d'émuler un Nunchuck, la manette d'extension de la WiiMote. La 4ème d'externaliser la caméra infra-rouge.

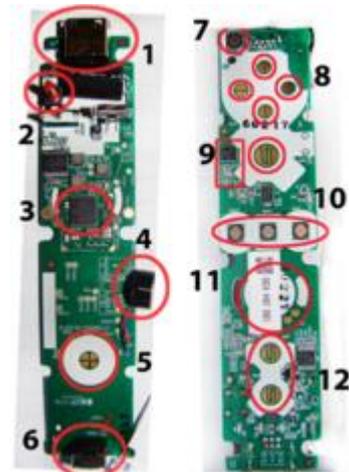
La manette

Pour démonter la WiiMote, un tournevis spécial est recommandé, les vis étant de la forme d'une étoile à 3 branches. Il est cependant possible de s'en sortir à l'aide d'un tournevis cruciforme relativement pointu ou plat.

2 encoches empêchent l'ouverture de la WiiMote du côté de la caméra IR, celle-ci se trouve au niveau des touches gauches et droites de la croix directionnelle.

Des vidéos sont disponibles montrant comment démonter une WiiMote. Après avoir démonté la WiiMote et analysé l'intérieur de celle-ci, On peut voir les éléments suivants :

- 1 Prise du nunchuck
- 2 Bouton de synchronisation
- 3 Puce Bluetooth
- 4 Vibreur
- 5 Bouton 'B'
- 6 Caméra IR
- 7 Bouton power
- 8 Croix directionnelle
- 9 Accéléromètre
- 10 Boutons '-' 'Home' '+'
- 11 Le haut-parleur
- 12 Boutons '1' et '2'



Étape 1 le contrôle des boutons

Des tests réalisés à l'aide d'un multimètre ont permis d'établir les informations suivantes au niveau des boutons:

Les boutons ont une masse commune ainsi qu'un connecteur dont la tension est de 3.16V. La masse est quant à elle relié à la borne négative de la batterie. J'ai donc choisi d'utiliser les sorties digitales de l'Arduino afin de piloter ces boutons. Je brancherai un diviseur de tension afin d'abaisser la tension à 3.16V. A la sortie de mon diviseur, je brancherai sur la borne '+' des boutons

L'Arduino impose une tension de 0-5V en sortie.
On prendra donc $R_2=2.2\text{kohms}$ et $R_1= 1.2\text{kohms}$

Suite à l'utilisation d'un Arduino DUE, un pont diviseur n'est pas nécessaire puisque les sorties ont une tension 0-3.3V. L'Arduino DUE présente également l'avantage d'offrir des Digital to Analog Converter (DAC). Il sera donc plus facile de piloter les tensions x,y,z de l'accéléromètre.

Afin de pouvoir piloter les boutons correctement il est nécessaire de mettre en commun les masses de la WiiMote et de l'Arduino DUE. J'ai également mis des résistances entre la sortie de l'Arduino et les boutons afin de ne pas causer de court-circuit. J'ai soudé un fil sur la masse du bouton 'A', les boutons ayant une masse commune cela permet de relier toutes les masses.

Étape 2 l'accéléromètre

L'accéléromètre de la WiiMote fournit des tensions variant de 1.10V à 1.8V. Cela correspond à des valeurs pour le DAC de l'Arduino comprise entre 1350 et 1450. Nous avons enlevé l'accéléromètre de la WiiMote afin que celui-ci ne perturbe pas la commande. Nous avons ensuite soudé des fils sur la capacité de filtrage de l'accéléromètre.

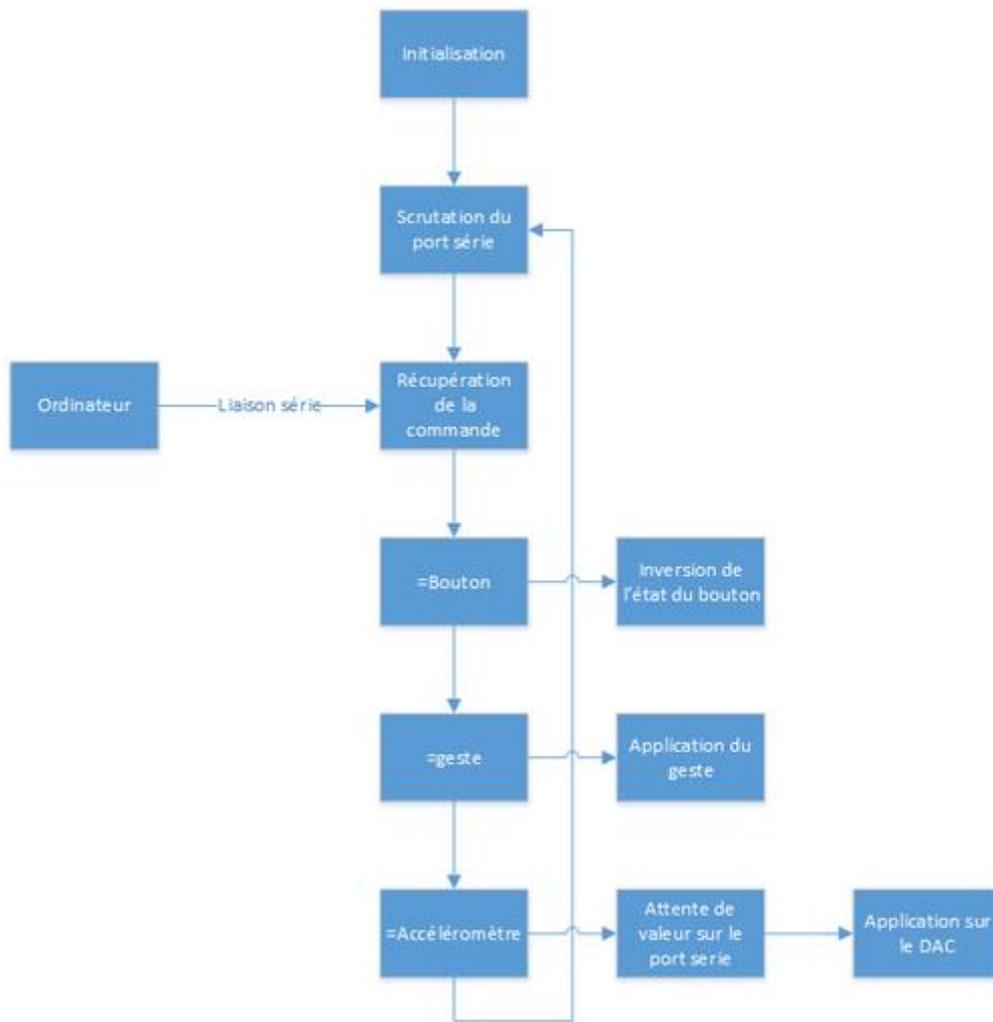
Étape 3 émulation du nunchuck

Par manque de temps cette étape n'a pas été traitée il est cependant possible de consulter des travaux sur l'émulation de nunchuck d'un autre groupe de projet de Polytech.

Le programme Arduino

Le programme Arduino fonctionne avec une communication série. On envoie les commandes via cette liaison. Les commandes sont listées ci-dessous.

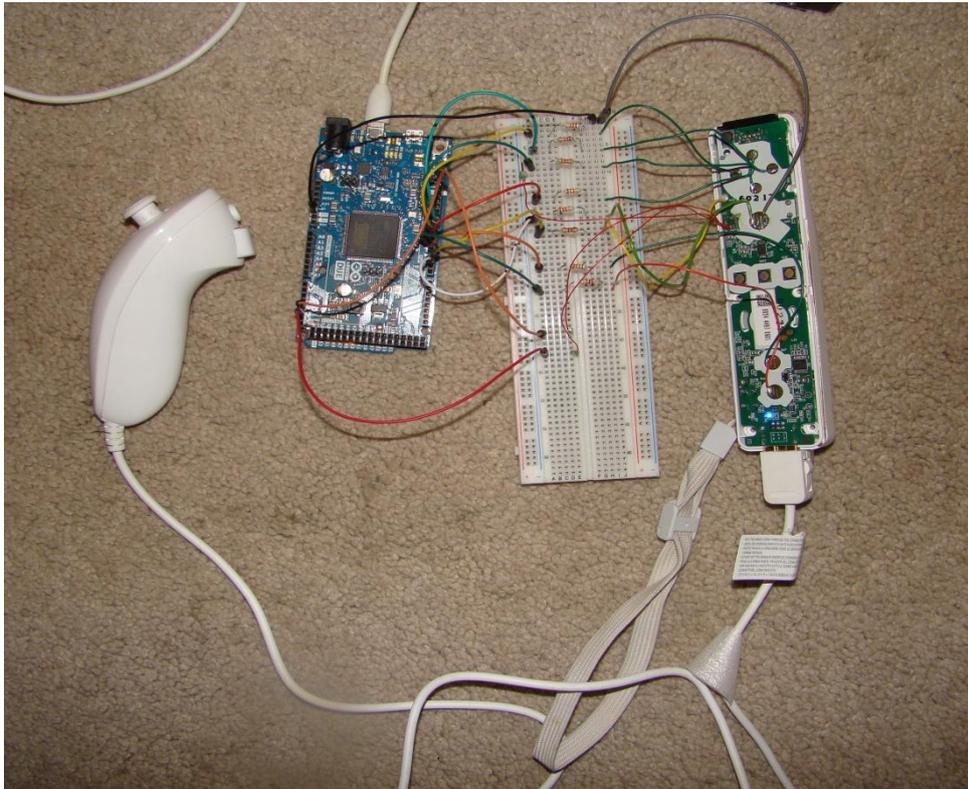
```
A inversion de l'état du bouton A
B ----- B
1 ----- 1
2 ----- 2
8 ----- flèche haut
4 ----- flèche gauche
5 ----- flèche bas
6 ----- flèche droite
y Attend la valeur d'accéléromètre puis applique la tension
x -----
z -----
p programme auto applique un geste (bowling)
```



Algorithme de l'Arduino

L'Arduino DUE ne possède que 2 DAC, ainsi seul 2 dimensions de l'accéléromètre peuvent être pilotées. Afin de régler ce problème, il est conseillé d'utiliser une PWM sur une sortie digitale de l'Arduino, suivit d'un filtre passe bas du type RC, avec $R=10\text{kohms}$ et $C=100\text{nF}$. Ce type de connexion n'a cependant pas été testé.

Le système final



Le système final se compose de 3 éléments.

- L'Arduino DUE
- Une breadboard
- La WiiMote

Les tests

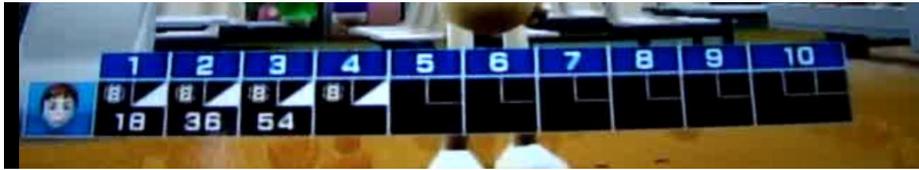
Les éducateurs utilisant le jeu de bowling du jeu WiiSport, il est logique de tester l'efficacité de la manette sur ce jeu. J'ai donc d'abord testé le pilotage des boutons. Le programme Arduino inverse l'état des boutons lors d'une réception de la commande associé. Ainsi pour simuler l'appui d'une touche il faut envoyer la commande de la touche 2 fois pour activer puis désactiver.

J'ai ensuite essayé de recréer un mouvement en piloter le Digital to Analog Converter. Pour donner l'illusion du mouvement, j'ai augmenté progressivement la tension de sortie du DAC.

L'algorithme est donc de la forme :

```
for(int i=1350 ; i<1450 ; i++){  
  
    analogWrite(DAC0,i) ;  
    delay(10);  
  
}
```

J'ai ensuite essayé de reproduire le mouvement de lancer. Le mouvement que j'ai créé rencontre une erreur lors du premier lancer et produit des spares aux lancers suivants.



8 quilles sont touchées au premier lancer puis les 2 dernières.



Conclusion

Même si l'idée de départ n'a pas pu aboutir (celle de créer un système émulant une WiiMote), l'objectif principal, de pouvoir piloter les boutons et l'accéléromètre a été rempli.

Le système asservit une WiiMote. L'asservissement consiste à avoir soudé des fils sur la WiiMote. Ils permettent de pouvoir contrôler l'activation des boutons ainsi que les valeurs de l'accéléromètre via un Arduino.

Des tests ont été effectués sur le jeu de bowling de WiiSport. Ce jeu est utilisé par les éducateurs dans leurs animations. Ces tests ont permis de mettre en place un exemple de mouvement, afin de démontrer l'asservissement de l'accéléromètre.

Un prototype de communication avec la Wii est donc terminé. Il faut donc maintenant travailler à la partie de détection des mouvements, pour que ce projet d'aide à la motricité de personnes handicapées puisse se concrétiser.