

James et la WiiMote

Fonctionne avec Linux, n'hésitez pas à nous envoyer la version windows

I. Connexion à la télécommande : WiiMote

Il est possible de récupérer les informations de la télécommande de la Wii. Nous allons voir ici comment piloter notre petit robot grâce à cette télécommande.

installer la librairie « libcwiiemote » : <http://libwiiemote.sourceforge.net/>
Sur ubuntu : dans Synaptic, rechercher wii, installer les paquets.

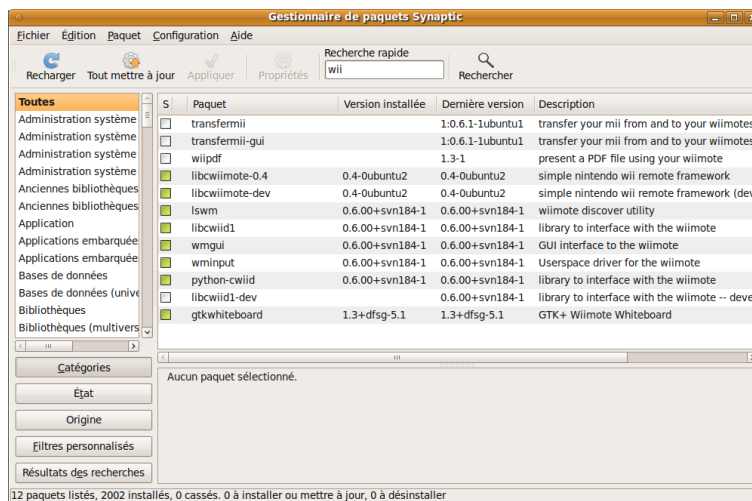


Illustration 1: Synaptic, installation des paquets concernant la Wii

Il est alors très facile de s'essayer à la récupération des informations de la wii en s'inspirant du code d'exemple fournit sur le site précédemment cité.

```

// print wiimote accelerometer data to stdout.
// replace xx:xx:xx:xx:xx:xx with the correct bluetooth address.

#include <stdio.h>
#include "wiimote_api.h"
int main()
{
    wiimote_t wi;
    wiimote_connect(&wi, "xx:xx:xx:xx:xx");
    wi.mode.acc = 1; // enable accelerometer
    while (wiimote_is_open(&wi)) {
        wiimote_update(&wi); // synchronize with wiimote
        if (wi.keys.home) { // check if home key is pressed
            wiimote_disconnect(&wi);
        }
        printf("x=%d y=%d z=%d\n", wi.axis.x, wi.axis.y, wi.axis.z);
    }
}

```

Il est ici demandé l'adresse mac de la wiimote. On peut aisément la connaître grace à la commande :

\$hcitool scan

Pour nous : 00:1F:C5:5C:50:3B

```

thomas@thomasPC: ~
Fichier  Édition  Affichage  Terminal  Aide
thomas@thomasPC:~$ hcitool scan
Scanning ...
    00:0B:CE:04:34:BB      Free2move WU
    00:1F:C5:5C:50:3B      Nintendo RVL-CNT-01
    00:0B:6B:5D:91:A0      Broadcom BCM2035
thomas@thomasPC:~$ █

```

Illustration 2: hcitool scan : liste des périphériques bluetooth

II. Connexion au module free2move

On a pu relever l'adresse mac de notre module bluetooth<-> uart grâce à la commande précédente.(00:0B:CE:04:34:BB)

Voici un petit script, qu'il est intéressant de placer dans /usr/bin/ afin de pouvoir l'exécuter depuis n'importe où dans la console.

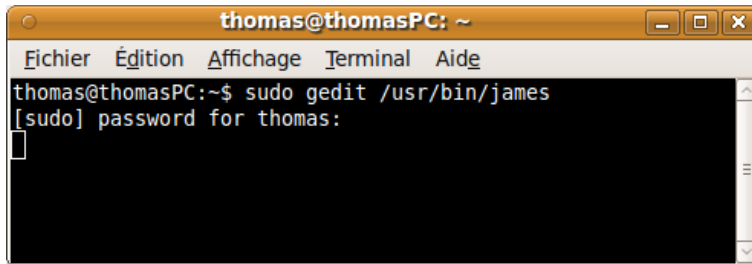


Illustration 3: Edition du script de connexion bluetooth

le script est le suivant : il faut remplacer l'adresse mac 00:0B:CE:02:40:B6 par la votre.

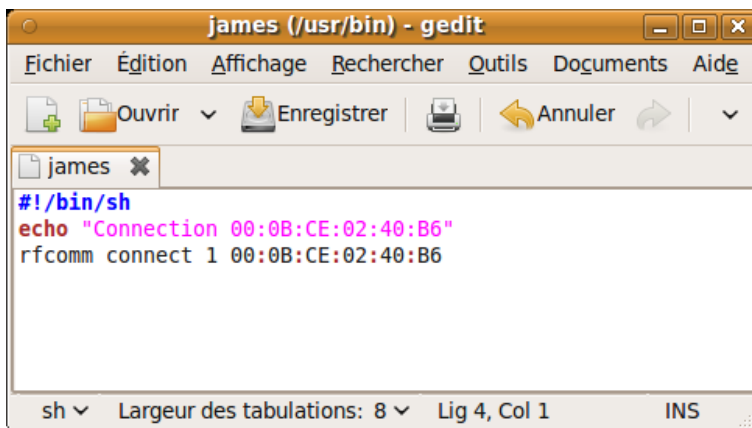


Illustration 4: Edition du script de connexion bluetooth 2

L'exécution du script permet alors de créer un port Com « rfcomm1 » attaché au module free2move.

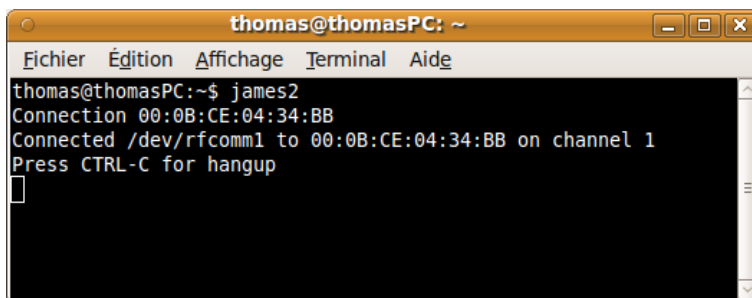


Illustration 5: Connexion au module Free2Move

III. Le programme

1. Utilisation du port Com :

On utilisera les fichiers « serie.h » et « serie.c » et notamment les fonctions suivantes :

```
/**
 * jamesWriteString : Send a string through the serial port.
 * @param parPtrString : string to be sent.
 * @return wn : number of byte written.
 */
int jamesWriteString( char * parPtrString);

/**
 * jamesWriteString : Send a cmd
 * @param parCmd : byte array
 * @param parLength : longueur
 * @return wn : number of byte written.
 */
int jamesWriteCmd(unsigned char * parCmd, int parLength);

/**
 * jamesOpenPort : open the serial port describe by james_SERIAL_PORT
 * @param none
 * @return mFd : -1 if fail to open.
 */
int jamesOpenPort(void);

/**
 * jamesRead : poll the serial port
 * @param parPtr : name of the thread, could be everything.
 * @return should never return...
 */
void * jamesRead(void * parPtr);
```

- La première « jamesWriteString » est utilisée pour écrire des données au format texte.
- La seconde « jamesWriteCmd » envoie une commande pas forcément texte : il faut préciser le nombre de caractères à envoyer.
- La troisième « jamesOpenPort » permet d'initialiser la communication série.
- La dernière « jamesRead » doit être lancée comme thread et ne se termine jamais.

```
/* Crée les thread */
if( pthread_create( &wThSerie, NULL, jamesRead, (void*)"James serial" ) < 0 )
{
    printf("Fail to create Thread : James serial \n");
    exit(-1);
}
[...]
/*Rejoint les threads */
(void)pthread_join( wThSerie, &wThreadRet );
```

2. Programme global

Pour ce qui est du reste, voici globalement comment fonctionne le programme, libre à vous de creuser un peu le code.

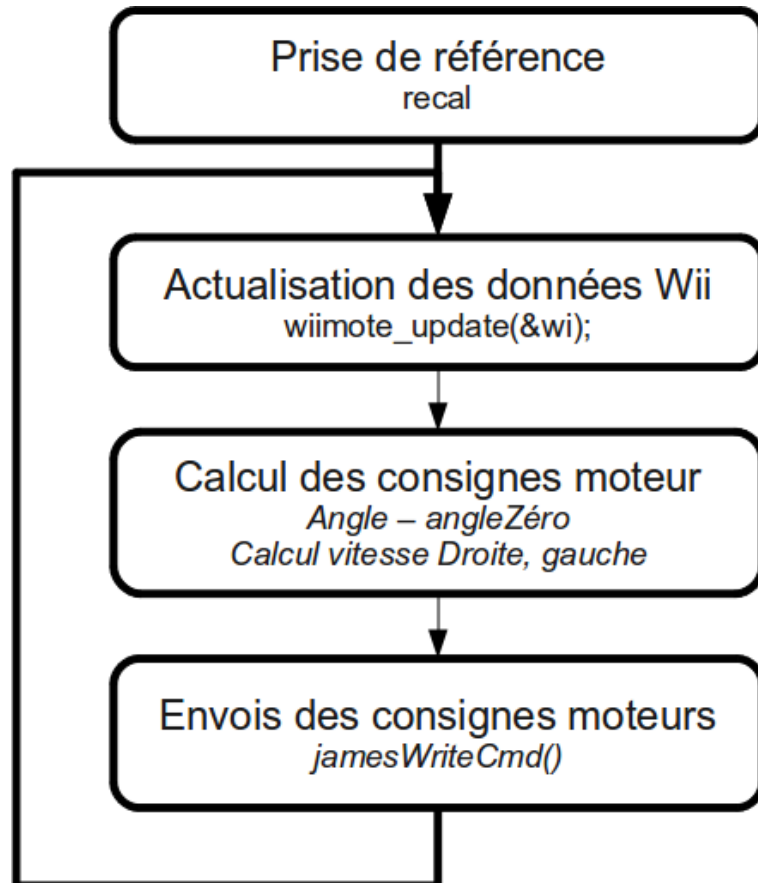


Illustration 6: Algorithme global

IV. Exécution du programme

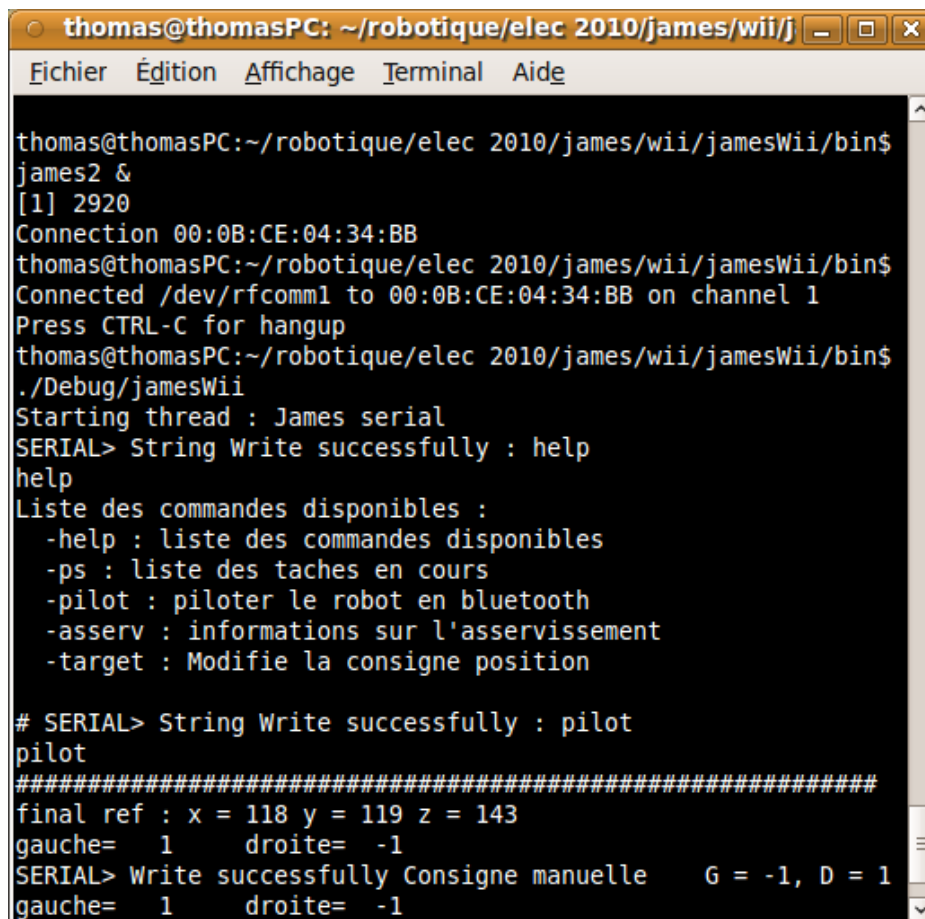
Il faut d'abord lancer la connexion serie bluetooth

```
$james2 &
```

puis se placer dans le répertoire contenant l'exécutable de notre programme, (ou l'appeler avec le bon chemin) et l'exécuter.

```
./jamesWii
```

voici un aperçu de chez nous :



```
thomas@thomasPC: ~/robotique/elec 2010/james/wii/jamesWii/bin$
james2 &
[1] 2920
Connection 00:0B:CE:04:34:BB
thomas@thomasPC:~/robotique/elec 2010/james/wii/jamesWii/bin$
Connected /dev/rfcomm1 to 00:0B:CE:04:34:BB on channel 1
Press CTRL-C for hangup
thomas@thomasPC:~/robotique/elec 2010/james/wii/jamesWii/bin$
./Debug/jamesWii
Starting thread : James serial
SERIAL> String Write successfully : help
help
Liste des commandes disponibles :
  -help : liste des commandes disponibles
  -ps : liste des taches en cours
  -pilot : piloter le robot en bluetooth
  -asserv : informations sur l'asservissement
  -target : Modifie la consigne position

# SERIAL> String Write successfully : pilot
pilot
#####
final ref : x = 118 y = 119 z = 143
gauche= 1    droite= -1
SERIAL> Write successfully Consigne manuelle    G = -1, D = 1
gauche= 1    droite= -1
```

Illustration 7: Résultat

Index des illustrations

Illustration 1: Synaptic, installation des paquets concernant la Wii.....	1
Illustration 2: hcitool scan : liste des périphériques bluetooth.....	2
Illustration 3: Edition du script de connexion bluetooth.....	2
Illustration 4: Edition du script de connexion bluetooth 2.....	2
Illustration 5: Connexion au module Free2Move.....	3
Illustration 6: Algorithme global.....	4
Illustration 7: Résultat.....	5