

Periféricos y Dispositivos de Interfaz Humana

Proyecto Wiimote




**UNIVERSIDAD
DE GRANADA**

Luis Miguel López López

1.- Funcionamiento del Wiimote

El wiimote consta de 12 botones en total, los 4 de la cruceta, A, B, 1, 2, los dos botones de volumen, el botón de encendido y el botón del menú home. Adicionalmente donde se encuentra la tapa hay un botón que es el de la conexión bluetooth.

Aparte de los botones básicos, el mando cuenta con una serie de acelerómetros que le permiten conocer su desplazamiento en los tres ejes. También cuenta con un sensor de infrarrojos que nos permite movernos en la pantalla “apuntando” hacia el emisor. Esto funciona con cualquier tipo de emisor infrarrojos siempre que tenga dos puntos de partida, como puede ser el emisor que viene con la consola o, en el caso de no disponer de él por algún motivo, dos velas separadas apropiadamente como podemos ver en el siguiente vídeo:

 [Swapping the Wii Sensor Bar with Candles \(it works!\)](#)

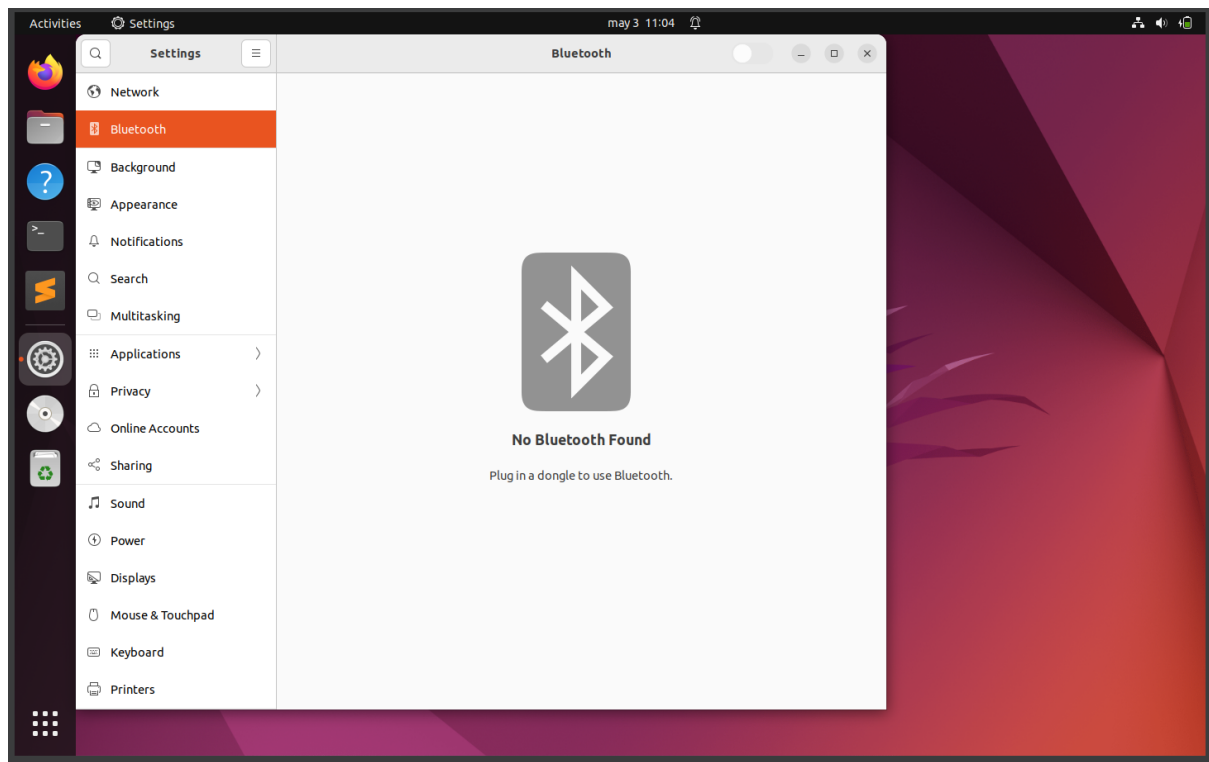
Adicionalmente algunos mandos más modernos cuentan con la tecnología MotionPlus que añade un movimiento más preciso tanto para acelerómetros como para infrarrojos.

Por último también cuenta con un altavoz, un motor interno que hace que el mando vibre y un puerto para conectar otros periféricos en la parte inferior.

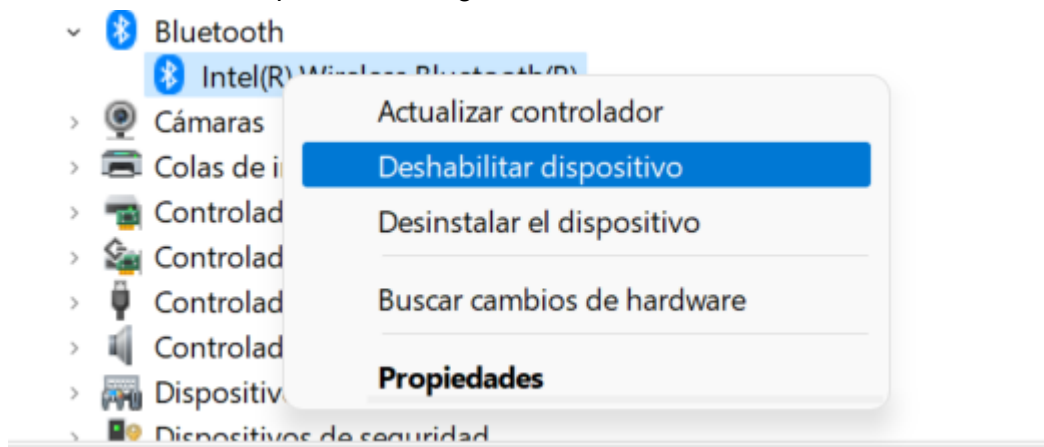
2.- Instalación de los módulos kernel.

Primero vamos a aclarar algunas consideraciones. El sistema utilizado es un Ubuntu 16.04 virtualizado con VirtualBox en un sistema Windows 11 como anfitrión.

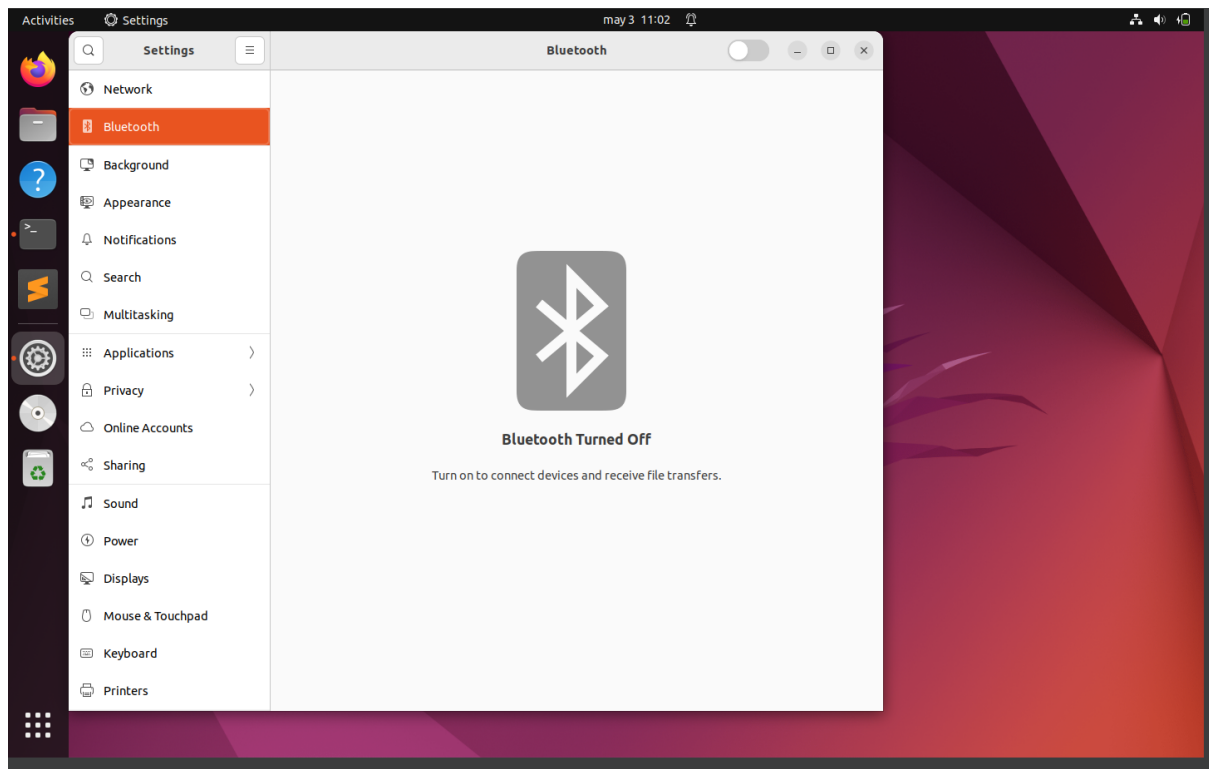
El primer paso es comprobar que podemos utilizar bluetooth en nuestra máquina virtual. Si Ubuntu no reconoce el bluetooth integrado en el ordenador:



Nos dirigiremos a administrador de dispositivos si tenemos un host windows y ahí deshabilitamos el dispositivo encargado del funcionamiento del bluetooth:



Volvemos a encender la máquina virtual y ahora tenemos que volver a habilitar el dispositivo de bluetooth y ya nos aparece que tenemos bluetooth en la máquina anfitrión, haciendo pruebas hemos observado que a veces desaparece el dispositivo de bluetooth por completo del anfitrión, unas veces ha funcionado directamente en la máquina virtual y otras veces hemos tenido que cerrarla e incluso reiniciar el anfitrión para que volviese a salir:



Comando que nos verifica que tenemos disponible el uso de bluetooth:
 hciconfig -a:

```

luismlpz@luismlpz-VirtualBox:~$ hciconfig -a
hci0: Type: Primary Bus: USB
      BD Address: 18:56:80:AD:3B:11 ACL MTU: 1021:4 SCO MTU: 96:6
      UP RUNNING
      RX bytes:1030 acl:0 sco:0 events:80 errors:0
      TX bytes:5452 acl:0 sco:0 commands:80 errors:0
      Features: 0xbf 0xfe 0x0f 0xfe 0xdb 0xff 0x7b 0x87
      Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
      Link policy: RSWITCH SNIFF
      Link mode: PERIPHERAL ACCEPT
      Name: 'luismlpz-VirtualBox'
      Class: 0x7c0000
      Service Classes: Rendering, Capturing, Object Transfer, Audio, Telephony
      Device Class: Miscellaneous,
      HCI Version: 5.1 (0xa) Revision: 0x100
      LMP Version: 5.1 (0xa) Subversion: 0x100
      Manufacturer: Intel Corp. (2)
  
```

Los siguientes pasos serán instalar xwiimote y el módulo del kernel hid-wiimote que ya forma parte del kernel de linux pero no se encuentra activado.

Para xwiimote clonamos el siguiente repositorio de github, ejecutamos el autorun y posteriormente el archivo configure (es posible que primero haya que darle permisos de ejecución, lo haremos con chmod).

Repo: <https://github.com/xwiimote/xwiimote>

El siguiente paso es instalar el paquete necesario para detectar el wiimote por bluetooth que será bluez y por último libudev-dev. Una vez instalado abrimos bluetoothctl y en la consola ejecutamos scan on y debería aparecer un dispositivo de Nintendo. Si escribimos devices

nos saldrán listados todos los dispositivos con sus direcciones MAC. Emparejamos con pair [Dirección MAC], trust [Dirección MAC] y connect [dirección MAC]:

```
[bluetooth]# scan on
Discovery started
[CHG] Controller 18:56:80:AD:3B:11 Discovering: yes
[CHG] Device 91:05:12:00:01:29 RSSI: -72
[CHG] Device 4C:BC:E9:7D:83:E4 RSSI: -83
[NEW] Device 00:1E:A9:3B:BD:A2 Nintendo RVL-CNT-01
[CHG] Device 4C:BC:E9:7D:83:E4 RSSI: -91
[bluetooth]# pair 00:1E:A9:3B:BD:A2
Attempting to pair with 00:1E:A9:3B:BD:A2
[CHG] Device 00:1E:A9:3B:BD:A2 Connected: yes
[CHG] Device 00:1E:A9:3B:BD:A2 Modalias: usb:v057Ep0306d0600
[CHG] Device 00:1E:A9:3B:BD:A2 UUIDs: 00001000-0000-1000-8000-00805f9b34fb
[CHG] Device 00:1E:A9:3B:BD:A2 UUIDs: 00001124-0000-1000-8000-00805f9b34fb
[CHG] Device 00:1E:A9:3B:BD:A2 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 00:1E:A9:3B:BD:A2 ServicesResolved: yes
[CHG] Device 00:1E:A9:3B:BD:A2 Paired: yes
Pairing successful
[CHG] Device 00:1E:A9:3B:BD:A2 WakeAllowed: yes
[CHG] Device 00:1E:A9:3B:BD:A2 ServicesResolved: no
[CHG] Device 00:1E:A9:3B:BD:A2 Connected: no
[CHG] Device 91:05:12:00:01:29 RSSI: -61
[bluetooth]# trust 00:1E:A9:3B:BD:A2
[CHG] Device 00:1E:A9:3B:BD:A2 Trusted: yes
Changing 00:1E:A9:3B:BD:A2 trust succeeded
[bluetooth]# connect 00:1E:A9:3B:BD:A2
Attempting to connect to 00:1E:A9:3B:BD:A2
[CHG] Device 00:1E:A9:3B:BD:A2 Connected: yes
Connection successful
[CHG] Device 00:1E:A9:3B:BD:A2 ServicesResolved: yes
[CHG] Device 91:05:12:00:01:29 RSSI: -72
[CHG] Device 4C:BC:E9:7D:83:E4 RSSI: -82
[CHG] Device 91:05:12:00:01:29 RSSI: -62
[CHG] Device 4C:BC:E9:7D:83:E4 RSSI: -90
[Nintendo RVL-CNT-01]#
```

Una vez conectado con xwiishow 1 podemos ver si funcionan los botones, con xwiishow -h podemos ver también como podemos modificar los leds del mando y activar y desactivar la vibración:

```
luismlpz@luismlpz-VirtualBox: ~  
- Keys -----+-----+ +-----+ +-----+  
    +-+      |   |       Accel x: -123 y: -27 z: -13 | XWIIMOTE SHOW |  
     | |      +-----+ +-----+  
    +-+ +-+   | IR #1: N/A x N/A #2: N/A x N/A #3: N/A x N/A #4: N/A x N/A |  
     | |       +-----+ +-----+  
    +-+ +-+   | MP x:          0 y:           0 z:            0 | LED (#1) -2  -3  -4 |  
     | |       +-----+ +-----+  
    +-+        Battery: 18% |##                | Ext: none              |||  
                                     +-----+ +-----+  
Device: gen10  
+-----+ +-----+  
  
( ) |         | ( )  
  
++++++  
+   ++  
+   ++  
++++++  
  
||  
||
```

+-----+ +-----+

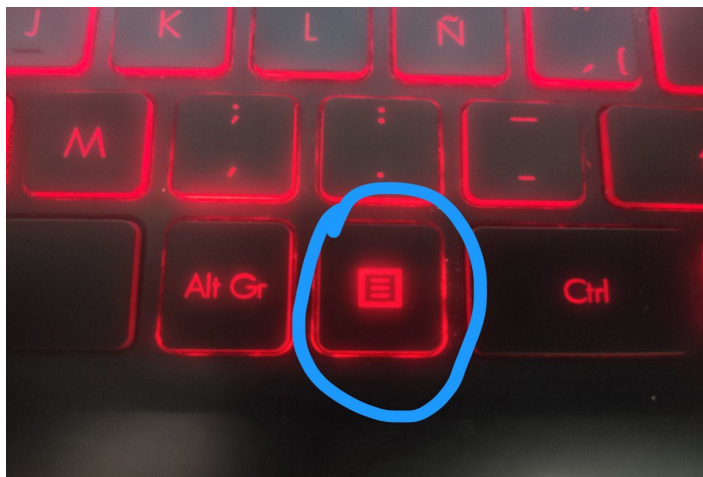
| Warning: Please run as root! (sysfs+evdev access needed) |

+-----+ +-----+

| Info: Screen smaller than 160x40: limited view

Una vez comprobamos que funcionan los botones podemos pasar a comprobar que funcionan tanto acelerómetros como infrarrojos. Podemos ver claramente en la línea Accel que funcionan correctamente los acelerómetros. Con los infrarrojos no es tan claro pero en el caso de no funcionar se vería como en la línea MP, es decir, sin ningún valor, en vez de con N/A. que significa que está en modo recepción pero que sí está funcionando. En una prueba posterior hemos visto como cambia el valor de IR #1 correctamente una vez conectado el emisor de infrarrojos de la consola original y ejecutado el programa como superusuario.

Sin embargo a la hora de probar la configuración que viene por defecto en el escritorio al menos a mí solo funcionaban la cruceta como si fuesen las flechas de dirección y los botones 1 y 2 como si fuesen el clic izquierdo y clic derecho del ratón, o incluso como si fuese enter y el botón del teclado que simula una pulsación de clic derecho del ratón.



Para cambiar la configuración podríamos usar el paquete `xf86-input-xwiimote` que podemos descargar del siguiente repositorio:

<https://github.com/xwiimote/xf86-input-xwiimote>

Seguiremos las instrucciones que pone en el readme, posiblemente haya que instalar algunos paquetes previos los cuales en nuestro caso han sido `ncurses-dev`, `libudev-dev` y `xserver-xorg-dev`, aunque si faltase otra librería al ejecutar el `autorun` o el `makefile` nos dirá cual es.

Una vez completada la instalación y copiado el archivo donde nos dice el readme allí podremos configurar el mapeado de los botones con las opciones que podemos ver en el `man xf86-input-xwiimote` o en la siguiente página:

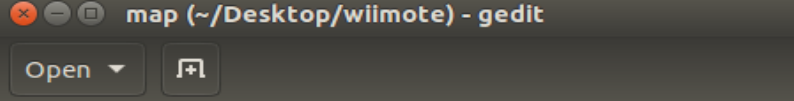
<https://manpages.ubuntu.com/manpages/impish/man4/xorg-xwiimote.4.html>

En nuestro caso hemos probado en la carpeta `/etc/X11/xorg.conf.d/` la cual he tenido que crear yo y en la carpeta `/usr/share/X11/xorg.conf.d/` que es la que se creó con la instalación, pero de ninguno de los dos modos hemos conseguido que funcione, algunas veces no cambiaba el mapeado y en otras directamente dejaban de funcionar los inputs que ya había previamente en el escritorio aunque `xwiishow` sí seguía reconociendo las pulsaciones de botón.

Alternativamente hemos encontrado otro conjunto de paquetes que sí nos permite mapear las teclas aunque no se ve que funcionen los acelerómetros y el infrarrojo no se puede probar correctamente porque en este caso parece ser que conecta directamente con la consola ya que necesita estar encendida para para que funcione el emisor.

Para este segundo método hemos empleado los paquetes `lswm` y `wminput`. Existe un ambiente gráfico pero no parece funcionar correctamente en ubuntu 16 llamado `wmgui`. Una vez instalados los paquetes usamos `gksudo gedit /etc/modules` y al final añadimos `uinput` para que arranque ese módulo al encender. Ahora podemos reiniciar la máquina o ejecutar `modprobe uinput`.

Para mapear los controles creamos un archivo de texto y escribimos lo siguiente dentro:



```
map (~/.Desktop/wiimote) - gedit
Open
Wiimote.A      =KEY_ENTER
Wiimote.B      =KEY_BACKSPACE
Wiimote.Up     =KEY_UP
Wiimote.Down   =KEY_DOWN
Wiimote.Left   =KEY_LEFT
Wiimote.Right  =KEY_RIGHT
Wiimote.1      =KEY_SPACE
Wiimote.2      =KEY_Q
Wiimote.Plus   =KEY_X
Wiimote.Minus  =KEY_C
Wiimote.Home   =KEY_MENU

Plugin.acc.X = ABS_X
Plugin.acc.Y = ABS_Y
|
```


Dependiendo de la versión o de la distribución KEY debe ser sustituido por BTN.
Ejecutamos la siguiente orden:

```
luismilpz@luismilpz-VirtualBox:~$ sudo wminput -w -c '/home/luismilpz/Desktop/wiimote/map'  
Put Wiimote in discoverable mode now (press 1+2)...  
Ready.  
luismilpz@luismilpz-VirtualBox:~$ sudo wminput -w -c '/home/luismilpz/Desktop/wiimote/map'
```

y aunque nos ponga presionar 1+2 hemos encontrado más fiable pulsar el botón trasero del mando en el compartimento de las baterías. Esto hace que el mando conecte directamente y apaga todas las luces del número de jugadores.

Si en el fichero de mapeo cambiamos acc por ir_ptr ese debería ser el plugin correcto para infrarrojos, para acelerómetro parece ser el que tenemos puesto ya que si no existiese saltaría un error al ejecutar el comando wminput.

3.- Instalación en Windows.

Primero tendremos que conectar el mando. Para ello nos dirigimos al panel de control y en Hardware y sonido clicamos en agregar dispositivo. Pulsamos el botón rojo del mando y esperamos a que se conecte, puede tardar varios intentos hasta que nos aparezca la siguiente pantalla que nos pedirá un PIN, pero simplemente podemos darle a siguiente y nos aparecerá en la lista de dispositivos conectados:

Dispositivos (3)



LUISM



Nintendo
RVL-CNT-01



Xbox Series X
Wired Controller
OPP Black

Impresoras (6)

Una vez conectado descargamos GlovePie y podemos usar uno de sus muchos scripts para controlar el mando como queramos e incluso utilizar los acelerómetros y los infrarrojos. Hemos probado el script de acelerómetros y la verdad que es un poco basto de controlar y hay que ir cambiando parámetros para que esté bien calibrado pero funciona.