

Dispositivos Bluetooth

Existen en RetroFive muchos sistemas que fueron pensados para ser jugados con un Gamepad y no con un joystick. Es por eso que mediante un dongle usb bluetooth podremos conectar diversos mandos inalámbricos para ser usados en el sistema y jugar de una manera más fiel y cercana.

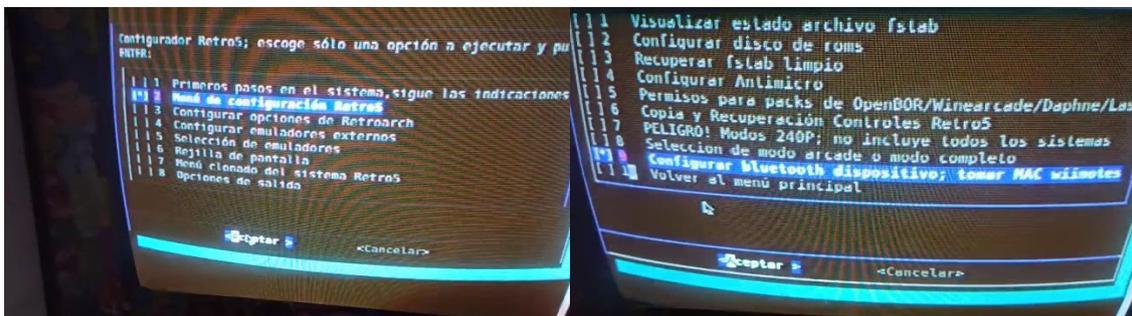
Necesitaremos uno o varios mandos como los de las consolas de la actual generación (Xbox One, PS4 o Switch), originales o no.



También necesitaremos de un dongle bluetooth por USB, en principio la inmensa mayoría de los dispositivos que podamos conectar serán compatibles, pero no hay una lista de compatibilidad como tal. A partir del protocolo Bluetooth 4.0 suele ser suficiente.

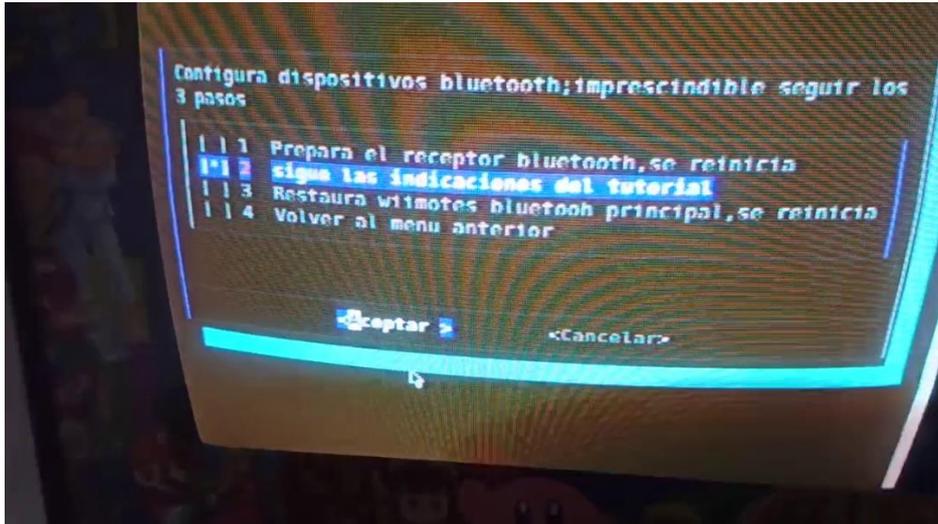


Necesitaremos disponer de un teclado para seguir el tutorial. Accederemos al configurador del escritorio R5 Menú, y accederemos a la opción para configurar nuestro dispositivo bluetooth (opción 2 “Menú de configuración Retro5”, opción 9 “Configurar bluetooth dispositivo; tomar MAC wiimotes”).

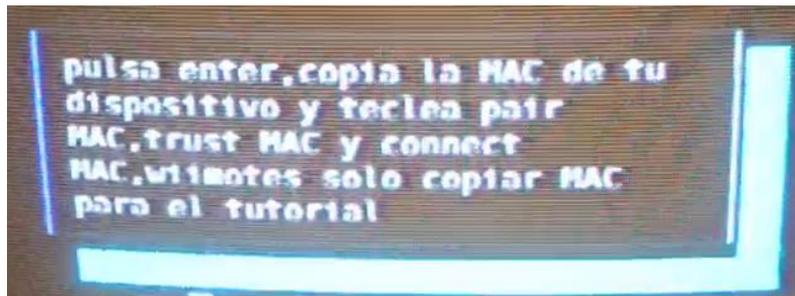


En este submenú, tenemos tres pasos: el primero prepara nuestro dongle usb bluetooth para permitir la conexión del gamepad (se reiniciará el pc y tendremos que volver a acceder al mismo submenú nuevamente); el segundo, nos abrirá una ventana de terminal donde deberemos introducir distintos comandos para averiguar la MAC de nuestro gamepad y emparejarlo con el sistema; el tercero restaurará, de tenerla, la configuración de nuestros

wiimotes, que más adelante se explicará como conectarlos para usar los sistemas con juegos de pistolas. También se reiniciará.



Nos centraremos en los comandos a introducir en la ventana de terminal, el segundo paso. En todo momento, el Gamepad debe de estar en modo “Discovery”; si por tiempo se fuera apagando, se vuelve a encender del mismo modo!!



Scan on: Nos activará la búsqueda de nuestro mando; el mando debe estar en modo “Discovery”; en Xbox One basta con pulsar el botón del logo XBOX; en PS4 es manteniendo unos segundos los botones “Share” y el logo de PS; la luz del mando parpadeará distinto a como si sólo pulsáramos el logo de PS. Si ha salido bien, veremos la MAC de nuestro Gamepad. Si diese error, introducir “Power on” y después el comando “Scan on” nuevamente.

Pair MAC: Tecleando el comando seguido de la MAC del Gamepad, emparejaremos el receptor y el mando. Debe de reportar conexión exitosa.

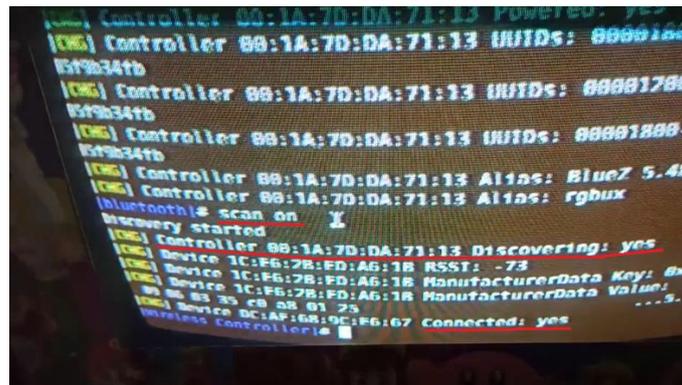
Trust MAC: Confirma que el dispositivo emparejado es el correcto; si pidiera una contraseña, esta es 0000.

Connect MAC: Fija el recuerdo para ambos dispositivos para cada reinicio del sistema. En ese momento, el mando debe fijar su luz y terminar con el modo “Discovery”.

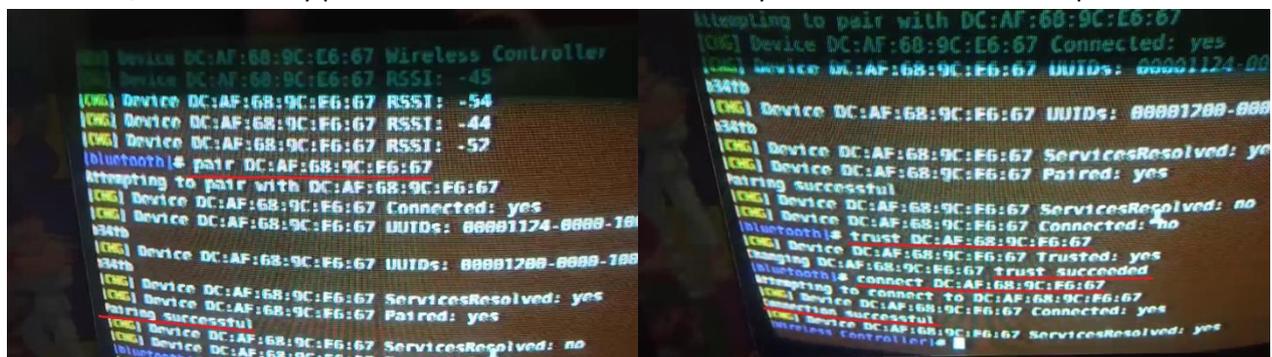
Saldremos del terminal escribiendo controlx o pulsando Alt+F4; recordad que hay que seleccionar la tercera opción para reiniciar el sistema y habilitar los wiimotes de haberse emparejado con anterioridad a este tutorial.

Vamos a verlo de manera más visual, con capturas de pantalla:

Usaremos el comando “Scan on”, siempre con el Gamepad en modo “Discovery”. Si lo detecta, veremos los mismos mensajes que en la captura siguiente.



Sin perder de vista que el mando siga en modo “Discovery”, ejecutaremos los comandos “Pair” MAC, “Trust” MAC y por último “Connect” MAC. Lo subrayado mostrará el éxito del proceso.



Si hubiera algún problema y al reiniciar con el último paso del menú (opción 3 “Restaura wiimotes bluetooth principal”), no reconociera Emulation Station nuestro Gamepad, seguid los siguientes comandos en la misma terminal donde se intentó anteriormente:

Devices: Si en el proceso nos devuelve algún error, se deberá seguramente a que en algún momento el Gamepad ha salido del modo “Discovery”. Este comando lista nuestros dispositivos bluetooth como el Gamepad y su MAC.

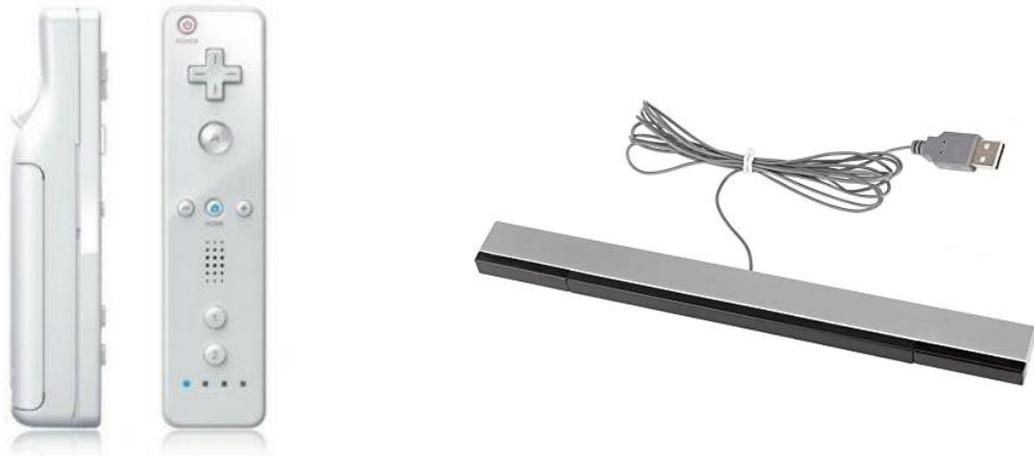
Remove MAC: Si diera problema al emparejarlo y para poder empezar el tutorial nuevamente, introducir el comando “Remove” y la MAC para eliminar el dispositivo, donde la MAC es en mi caso DC:AF:68:9C:E6:67.

Power on: Fuerza el reseteo del dongle usb bluetooth y nos permite seguir el tutorial anterior, con los comandos en orden: “Scan on”, “Pair MAC”, “Trust MAC” y “Connect MAC”, siempre con el Gamepad en modo “Discovery”!!!

Introducción wiimotes

Los wiimotes nos sirven en el sistema RetroFive para poder jugar de una manera más fiel y cercana a los juegos denominados “de pistola” de diversos sistemas arcade y domésticos, pulsando los botones 1 y 2 de cada mando en Emulation Station para activarlos y usarlos.

Necesitaremos un par de wiimotes, a poder ser originales de Nintendo, y una barra receptora de infrarrojos por USB o inalámbrica a pilas que se pueden comprar por Aliexpress, Ebay, Amazon, etc. La barra original que venía con la Wii nos vale, pero en cuanto pulsemos algún botón de un Wiimote la consola se encenderá, ya que la barra recibe la alimentación de la propia Wii, y esta debe de estar encendida para que funcione. No compensa, lógicamente.



También necesitaremos de un dongle bluetooth por USB, en principio la inmensa mayoría de los dispositivos que podamos conectar serán compatibles, pero no hay una lista de compatibilidad como tal. Los wiimotes funcionan con protocolo Bluetooth 2.0.

Bluetooth 4.0 2.4G 50M



En adición, una Dolphinbar nos servirá también y podremos prescindir de un dongle bluetooth al ya disponer de él la barra. Puede que poniendo el modo 1 o el 2 de la barra nos reconozca automáticamente los wiimotes al sincronizarlos, sin necesidad de seguir el tutorial.

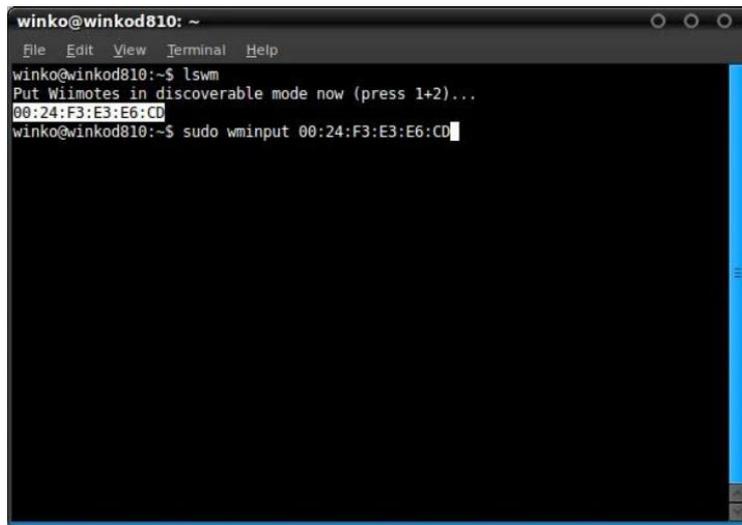
En Emulation Station, en “Extras”, deberemos de activar la función de los wiimotes. Por optimización del sistema, por defecto vienen desactivados. Cuando no vayamos a usarlos, conviene desactivar la función en el mismo menú.



Identificación de las MAC de los Wiimotes

Al ser el sistema RetroFive una imagen clonada, tendremos unas MAC que no coincidirán con las de nuestros Wiimotes.

Si el método anterior para emparejar Gamepads bluetooth no nos sirve para identificar nuestros Wiimotes, con el dongle bluetooth conectado al pc y desde el escritorio, abriremos una ventana de terminal y teclearemos **"lswm"**. En ese momento, deberemos de pulsar los botones 1 y 2 del primer Wiimote y aparecerá una dirección MAC. Con el comando **"sudo wminput xx:xx:xx:xx:xx:xx"** donde las "x" serán sustituidas por nuestra MAC, podremos comprobar el movimiento del cursor con el wiimote pulsando los botones 1 y 2 para sincronizar; no tiene más utilidad. Con Control+C finalizaremos la prueba del wiimote. Conviene no repetir los pasos del tutorial para identificar el segundo Wiimote hasta que tengamos terminado el proceso y nos funcione el primer Wiimote.

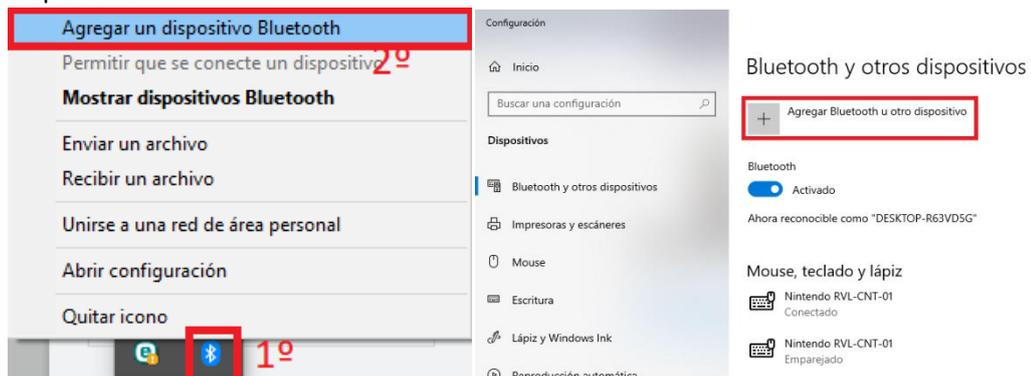


```
winko@winkod810: ~  
File Edit View Terminal Help  
winko@winkod810:~$ lswm  
Put Wiimotes in discoverable mode now (press 1+2)...  
00:24:F3:E3:E6:CD  
winko@winkod810:~$ sudo wminput 00:24:F3:E3:E6:CD
```

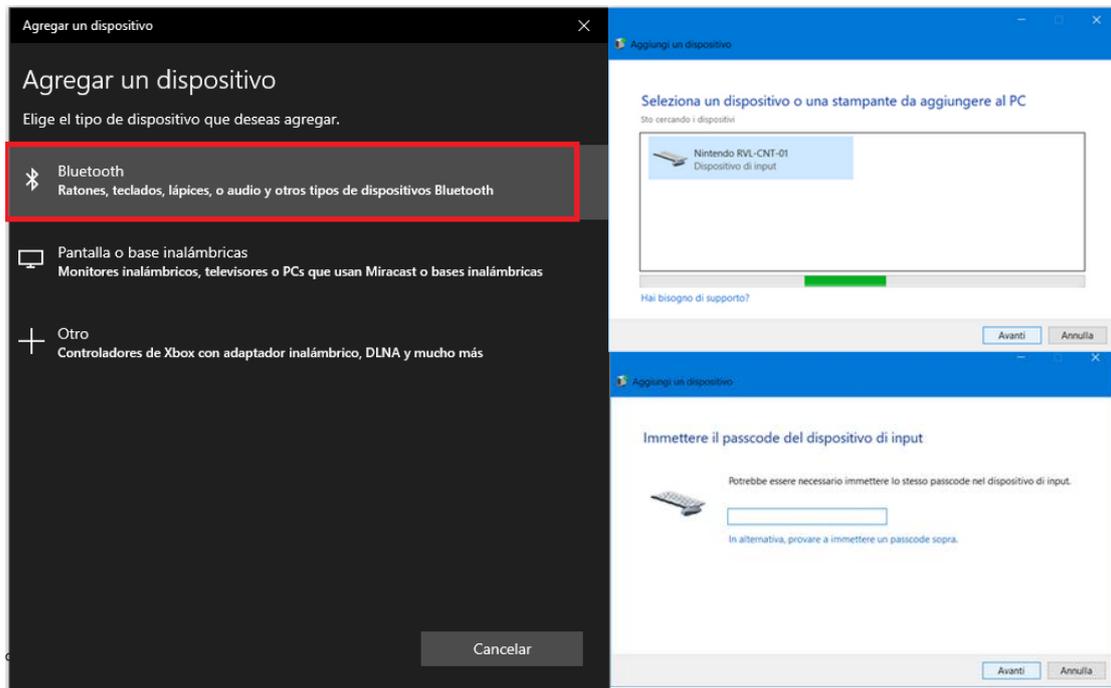
Si por alguna razón, el dongle bluetooth no respondiera o nos devolviera el mensaje "Bluetooth device inquiry error", verificamos que Linux detecta el adaptador Bluetooth, tecleando en el terminal **"hcitool dev"**. De ser negativa la respuesta, deberemos de probar con otro dongle bluetooth. De ser positiva la respuesta, podemos intentar que nos sirva para jugar, pero necesitaremos averiguar las MAC de nuestros Wiimote mediante otros métodos.

Si usamos Windows 10, seguid estos pasos:

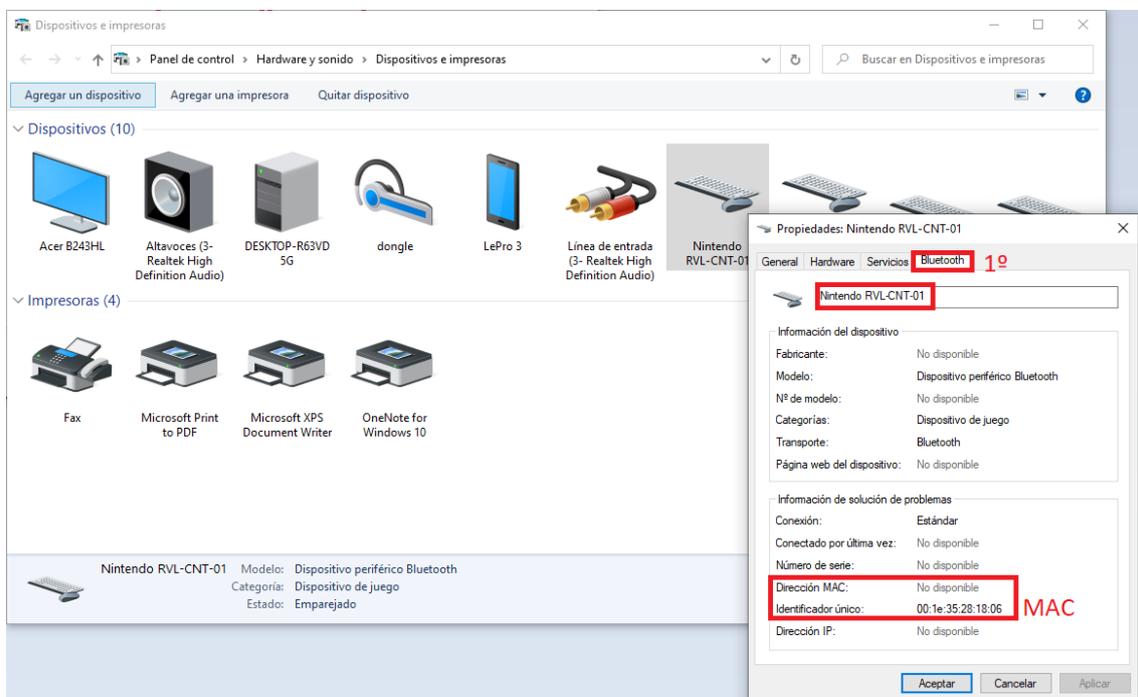
- En el icono de Bluetooth, hacemos clic derecho y seleccionamos "Agregar un dispositivo Bluetooth".
- En la siguiente ventana de configuración, seleccionamos "Agregar Bluetooth u otro dispositivo".



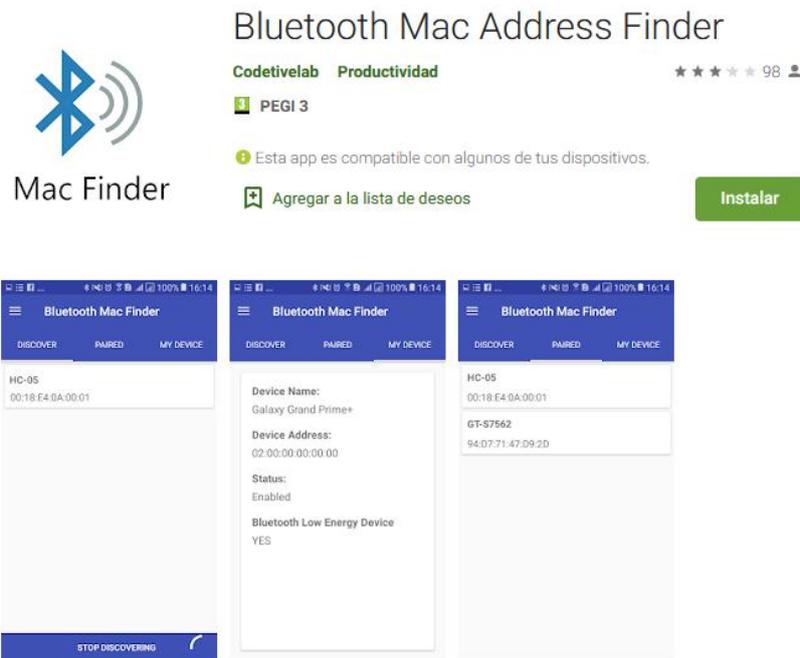
- En el tipo de dispositivo a agregar, seleccionaremos “Bluetooth”.
- En la siguiente ventana, deberemos de mantener pulsados los botones 1 y 2 del Wiimote mientras el sistema hace la búsqueda... al poco tiempo, aparecerá detectado como “Nintendo RVL-CNT-01”, lo seleccionamos y le damos a “siguiente”.



- Por último, abriremos el Panel de control/Hardware y sonido/Dispositivos e impresoras; en dispositivos, veremos nuestro Wiimote como “Nintendo RVL-CNT-01”; hacemos clic encima, y en la pestaña “Bluetooth” veremos la MAC como en la imagen.



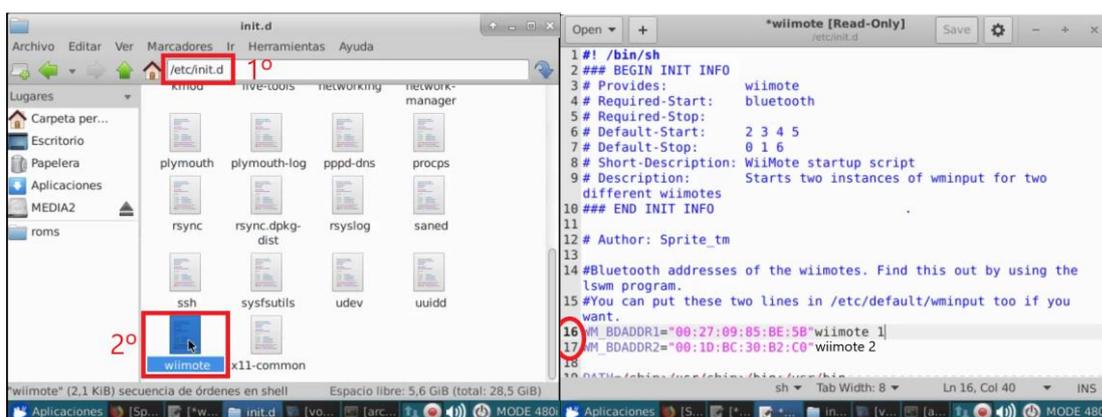
Si preferimos usar una app Android llamada “Bluetooth Address Finder” (hay varias opciones en Play Store), abierta la app, activamos el bluetooth y hacemos una búsqueda de dispositivos. Cuando la app lo indique, pondremos el primer Wiimote en “Discovery mode” con los botones 1 y 2. La dirección MAC debería mostrarse en la app. Lo mismo para el segundo Wiimote.



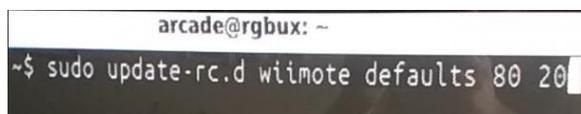
Conviene anotar las Mac de los Wiimotes a buen recaudo.

Cambiar las MAC de los Wiimotes en el sistema RetroFive

Introduciendo “**sudo thunar**” en el menú del escritorio “aplicaciones/ejecutar programa”, nos abre una ventana de “Administrador de archivos” pero con permisos de modificación; es importante saber qué se hace bajo este modo. Navegamos hacia la siguiente ruta: **etc/init.d/**. Dentro de esa carpeta, editaremos un archivo llamado “**wiimote**” (botón derecho, “Text Editor”). Las líneas 16 y 17 son las MAC originales de nuestro sistema, por orden, la línea 16 corresponde al primer Wiimote y la línea 17 corresponde al segundo Wiimote; ahí deberemos de copiar nuestras MAC obtenidas con alguno de los métodos explicados anteriormente. En este punto, de momento sólo sustituiremos la MAC del primer Wiimote. Guardaremos los cambios del archivo una vez modificada la MAC.

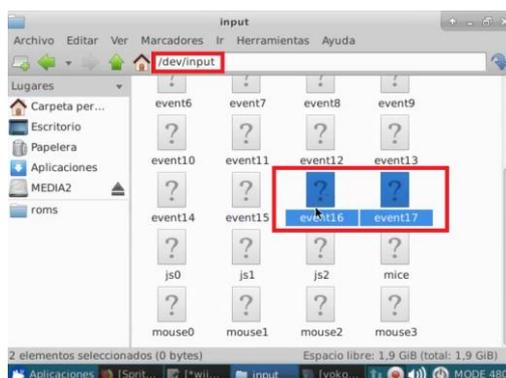


Ahora, hay que introducir en una ventana de terminal el comando “**sudo update-rc.d wiimote defaults 80 20**”. Sirve para colocar la MAC del archivo “**wiimote**” en el kernel al inicio.



Identificación de los archivos de control de los Wiimotes

En el menú del escritorio “aplicaciones/administrador de archivos” abriremos una ventana de “Administrador de archivos” de no conservar la anterior y accederemos a la siguiente ruta: **/dev/input**. Dentro encontraremos unos archivos denominados “**eventX**”, donde **X** será el número asignado por el sistema a nuestro wiimote para reconocerlo al arrancar el sistema. Durante la realización del tutorial, en nuestro caso los dos números más altos de los archivos “event” correspondían a nuestros dos wiimotes (16 y 17), aunque pueden perfectamente colocarse en uno o varios números inferiores y separados. Tomaremos el número 16 como ejemplo.



Para comprobar si el archivo “event” corresponde al Wiimote, abriremos una “Ventana de terminal” y ejecutaremos el siguiente comando: **evtest /dev/input/eventX**, donde sustituiremos la X con el número visto antes, en nuestro caso el 16. En ese momento, sincronizaremos el Wiimote con los botones 1 y 2 a la vez hasta encenderse el led 1. Podremos testear y probar la funcionalidad del Wiimote correspondiente a ese “event”, variando con el movimiento del mando o pulsación de botones lo mostrado en pantalla. Si con el ratón nos desplazamos hacia la parte superior, donde se lee **“Input device name”** debe aparecer **“Nintendo Wiimote”**. De no ser así, el “event” es erróneo y deberemos de probar con otro.

```

arcade@rgbux:~$ evtest /dev/input/event16
arcade@rgbux:~$ evtest /dev/input/event16
Input driver version is 1.0.1
Input device ID: bus 0x5 vendor 0x1 product 0x1 version 0x5
Input device name: "Nintendo Wiimote"
Supported events:
Event type 0 (EV_SYN)
Event type 1 (EV_KEY)
Event code 23 (KEY_I)
Event code 35 (KEY_H)
Event code 36 (KEY_J)
Event code 37 (KEY_K)
Event code 38 (KEY_L)
Event code 48 (KEY_B)
Event code 49 (KEY_N)
Event code 50 (KEY_M)
Event code 97 (KEY_RIGHTCTRL)
Event code 100 (KEY_RIGHTALT)
Event code 272 (BTN_LEFT)
Event type 3 (EV_ABS)
Event code 0 (ABS_X)
Value 0
Min 50
Max 974
Event code 1 (ABS_Y)
Value 0
Min 50
Max 718
Properties:
Testing ... (interrupt to exit)

```

Guardado de configuración de los Wiimotes

En la “Ventana de terminal”, ejecutaremos lo siguiente para editar el “xorg”: **“sudo mousepad /etc/X11/xorg.conf”**.

En nuestro caso, en las líneas 61 y 67 (primer y segundo Wiimote respectivamente), sustituiremos los números de los archivos “eventX” que aparecen por los comprobados antes; en el ejemplo, fueron los números 16 y 17, pero no tienen por qué coincidir o ser correlativos.

```

arcade@rgbux:~$ sudo gedit /etc/X11/xorg.conf
** (gedit:10763): WARNING **: 16:38:50.151: Set document metadata failed: Setting attribute metadata:gedit-position not supported
arcade@rgbux:~$ sudo gedit /etc/X11/xorg.conf

```

```

49 EndSubSection
50 EndSection
51 Section "ServerFlags"
52 Option "blank time" "0"
53 Option "standby time" "0"
54 Option "suspend time" "0"
55 Option "off time" "0"
56 Option "dpms" "false"
57 EndSection
58 Section "InputDevice"
59 Identifier "WiiMote0"
60 Driver "evdev"
61 Option "Device" "/dev/input/event16"
62 Option "SendCoreEvents" "True"
63 EndSection
64 Section "InputDevice"
65 Identifier "WiiMote1"
66 Driver "evdev"
67 Option "Device" "/dev/input/event17"
68 Option "SendCoreEvents" "True"
69 EndSection
70

```

Podemos **reiniciar** y empezar el tutorial de nuevo para añadir un segundo Wiimote al sistema.

Adición de un segundo ratón virtual

Para poder jugar a dobles en algunos juegos con dos wiimotes, debemos de seguir el siguiente procedimiento para añadir un segundo ratón virtual con el que el emulador podrá gestionar la posición de este segundo puntero como hace por defecto con el primer puntero.

Abriremos una ventana de terminal, y con el comando “*xinput list*” nos muestra los dispositivos disponibles para asociar a un teclado o puntero virtuales.

```
Terminal - arcade@Retro5: ~
arcade@Retro5:~$ xinput list
Virtual core pointer          id=2  [master poin
ter (3)]
├─ Virtual core XTEST pointer  id=4  [slave poin
ter (2)]
├─ MI Dongle MI Wireless Mouse id=11 [slave poin
ter (2)]
├─ MI Dongle MI Wireless Mouse id=12 [slave poin
ter (2)]
├─ Telink Wireless Receiver    id=14 [slave poin
ter (2)]
├─ Nintendo Wiimote           id=16 [slave poin
ter (2)]
├─ Nintendo Wiimote           id=17 [slave poin
ter (2)]
├─ Nintendo Wiimote           id=18 [slave poin
ter (2)]
└─ Virtual core keyboard      id=3  [master keyb
oard (2)]
├─ Virtual core XTEST keyboard id=5  [slave keyb
oard (3)]
├─ Power Button               id=6  [slave keyb
oard (3)]
└─ Video Bus                  id=7  [slave keyb
```

Debemos de fijarnos en la *id* de los Wiimotes asociados a “pointer”, no a “keyboard”. En el ejemplo, nos interesa la *id* 17 del segundo wiimote, no la *id* 23.

```
Terminal - arcade@Retro5: ~
├─ MI Dongle MI Wireless Mouse id=12 [slave poin
ter (2)]
├─ Telink Wireless Receiver    id=14 [slave poin
ter (2)]
├─ Nintendo Wiimote           id=16 [slave poin
ter (2)]
├─ Nintendo Wiimote           id=17 [slave poin
ter (2)]
├─ Nintendo Wiimote           id=18 [slave poin
ter (2)]
└─ Virtual core keyboard      id=3  [master keyb
oard (2)]
├─ Virtual core XTEST keyboard id=5  [slave keyb
oard (3)]
├─ Power Button               id=6  [slave keyb
oard (3)]
├─ Video Bus                  id=7  [slave keyb
oard (3)]
├─ Video Bus                  id=8  [slave keyb
oard (3)]
├─ Power Button               id=9  [slave keyb
oard (3)]
├─ MI Dongle MI Wireless Mouse id=10 [slave keyb
oard (3)]
├─ Telink Wireless Receiver    id=13 [slave keyb
oard (3)]
├─ HP WMI hotkeys             id=15 [slave keyb
oard (3)]
├─ MI Dongle MI Wireless Mouse id=19 [slave keyb
oard (3)]
├─ MI Dongle MI Wireless Mouse id=20 [slave keyb
oard (3)]
├─ Telink Wireless Receiver    id=21 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=22 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=23 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=24 [slave keyb
oard (3)]
└─ MI Dongle MI Wireless Mouse id=10 [slave keyb
```

Crearemos un controlador maestro nuevo para nuestro segundo ratón/wiimote con el comando “*xinput create-master wiimote2*”, donde “wiimote2” es el nombre escogido. Listaremos nuevamente con el comando “*xinput list*” para anotar su *id* (en el ejemplo, 25).

```
Terminal - arcade@Retro5: ~
├─ Video Bus                  id=8  [slave keyb
oard (3)]
├─ Power Button               id=9  [slave keyb
oard (3)]
├─ MI Dongle MI Wireless Mouse id=10 [slave keyb
oard (3)]
├─ Telink Wireless Receiver    id=13 [slave keyb
oard (3)]
├─ HP WMI hotkeys             id=15 [slave keyb
oard (3)]
├─ MI Dongle MI Wireless Mouse id=19 [slave keyb
oard (3)]
├─ MI Dongle MI Wireless Mouse id=20 [slave keyb
oard (3)]
├─ Telink Wireless Receiver    id=21 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=22 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=23 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=24 [slave keyb
oard (3)]
└─ MI Dongle MI Wireless Mouse id=10 [slave keyb
oard (3)]
arcade@Retro5:~$ xinput create-master wiimote2
arcade@Retro5:~$ xinput list
├─ HP WMI hotkeys             id=15 [slave keyb
oard (3)]
├─ MI Dongle MI Wireless Mouse id=19 [slave keyb
oard (3)]
├─ MI Dongle MI Wireless Mouse id=20 [slave keyb
oard (3)]
├─ Telink Wireless Receiver    id=21 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=22 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=23 [slave keyb
oard (3)]
├─ Nintendo Wiimote           id=24 [slave keyb
oard (3)]
├─ wiimote2 pointer          id=25 [master poin
ter (26)]
├─ wiimote2 XTEST pointer     id=27 [slave poin
ter (25)]
├─ wiimote2 keyboard          id=26 [master keyb
oard (25)]
├─ wiimote2 XTEST keyboard    id=28 [slave keyb
oard (26)]
└─ MI Dongle MI Wireless Mouse id=10 [slave keyb
```

Por último, asociaremos este controlador maestro a nuestro segundo wiimote con el comando **“xinput reattach 17 25”**, donde 17 es el id de nuestro segundo wiimote dentro de pointer y 25 es nuestro controlador maestro creado anteriormente. Una vez introducido, deberemos de tener dos punteros y controlarlos independientemente con cada wiimote.

```

Terminal - arcade@Retro5: ~
oard (3)]
└─ HP WMI hotkeys id=15 [slave keyb
oard (3)]
└─ MI Dongle MI Wireless Mouse id=19 [slave keyb
oard (3)]
└─ MI Dongle MI Wireless Mouse id=20 [slave keyb
oard (3)]
└─ Telink Wireless Receiver id=21 [slave keyb
oard (3)]
└─ Nintendo Wiimote id=22 [slave keyb
oard (3)]
└─ Nintendo Wiimote id=23 [slave keyb
oard (3)]
└─ Nintendo Wiimote id=24 [slave keyb
oard (3)]
└─ wiimote2 pointer id=25 [master poin
ter (26)]
└─ wiimote2 XTEST pointer id=27 [slave poin
ter (25)]
└─ wiimote2 keyboard id=26 [master keyb
oard (25)]
└─ wiimote2 XTEST keyboard id=28 [slave keyb
oard (26)]
arcade@Retro5:~$ xinput reattach 17 25

```

Para deshacer lo realizado en el tutorial, sólo debemos asociar nuevamente el wiimote2 al “pointer” original. Para ello, listamos dispositivos con el comando **“xinput list”**, nos fijamos en la id del puntero virtual original (en el ejemplo, id 2), y con el comando **“xinput reattach 17 2”** reasociaremos el wiimote2 con el valor de origen, volviendo a tener un solo puntero.

```

Terminal - arcade@Retro5: ~
arcade@Retro5:~$ si queremos volver al principio
si: orden no encontrada
arcade@Retro5:~$ xinput list
└─ virtual core pointer id=2 [master poin
ter (3)]
└─ Virtual core XTEST pointer id=4 [slave poin
ter (2)]
└─ MI Dongle MI Wireless Mouse id=11 [slave poin
ter (2)]
└─ MI Dongle MI Wireless Mouse id=12 [slave poin
ter (2)]
└─ Telink Wireless Receiver id=14 [slave poin
ter (2)]
└─ Nintendo Wiimote id=16 [slave poin
ter (2)]
└─ Nintendo Wiimote id=18 [slave poin
ter (2)]
└─ Virtual core keyboard id=3 [master keyb
oard (2)]
└─ Virtual core XTEST keyboard id=5 [slave keyb
oard (3)]
└─ Power Button id=6 [slave keyb

Terminal - arcade@Retro5: ~
ard (3)]
└─ MI Dongle MI Wireless Mouse id=19 [slave keybo
ard (3)]
└─ MI Dongle MI Wireless Mouse id=20 [slave keybo
ard (3)]
└─ Telink Wireless Receiver id=21 [slave keybo
ard (3)]
└─ Nintendo Wiimote id=22 [slave keybo
ard (3)]
└─ Nintendo Wiimote id=23 [slave keybo
ard (3)]
└─ Nintendo Wiimote id=24 [slave keybo
ard (3)]
└─ wiimote2 pointer id=25 [master point
er (26)]
└─ Nintendo Wiimote id=17 [slave point
er (25)]
└─ wiimote2 XTEST pointer id=27 [slave point
er (25)]
└─ wiimote2 keyboard id=26 [master keybo
ard (25)]
└─ wiimote2 XTEST keyboard id=28 [slave keybo
ard (26)]
arcade@Retro5:~$ xinput reattach 17 2

```

Conviene eliminar el controlador maestro creado anteriormente, con el comando **“xinput remove-master 25”**.

```

Terminal - arcade@Retro5: ~
ard (3)]
└─ MI Dongle MI Wireless Mouse id=20 [slave keybo
ard (3)]
└─ Telink Wireless Receiver id=21 [slave keybo
ard (3)]
└─ Nintendo Wiimote id=22 [slave keybo
ard (3)]
└─ Nintendo Wiimote id=23 [slave keybo
ard (3)]
└─ Nintendo Wiimote id=24 [slave keybo
ard (3)]
└─ wiimote2 pointer id=25 [master point
er (26)]
└─ Nintendo Wiimote id=17 [slave point
er (25)]
└─ wiimote2 XTEST pointer id=27 [slave point
er (25)]
└─ wiimote2 keyboard id=26 [master keybo
ard (25)]
└─ wiimote2 XTEST keyboard id=28 [slave keybo
ard (26)]
arcade@Retro5:~$ xinput reattach 17 2
arcade@Retro5:~$ xinput remove-master 25
arcade@Retro5:~$

```