

# *Vircon32*

## CONSOLA VIRTUAL DE 32 BITS



## Especificación del sistema

# Parte 8: Elementos externos

---

Documento con fecha 2024.01.17

Escrito por Carra

## ¿Qué es esto?

Este documento es la parte número 8 de la especificación del sistema Vircon32. Esta serie de documentos define el sistema Vircon32, y provee una especificación completa que describe en detalle sus características y comportamiento.

El principal objetivo de esta especificación es definir un estándar de lo que es un sistema Vircon32, y cómo debe implementarse un sistema de juego para que se considere conforme a él. Además, al ser Vircon32 es un sistema virtual, un importante objetivo adicional de estos documentos es proporcionar a cualquiera el conocimiento para crear sus propias implementaciones de Vircon32.

---

## Sobre Vircon32

El proyecto Vircon32 fue creado de forma independiente por Carra. El sistema Vircon32 y su material asociado (incluyendo documentos, software, código fuente, arte y cualquier otro elemento relacionado) son propiedad del autor original.

Vircon32 es un proyecto libre y de código abierto en un esfuerzo por promover que cualquiera pueda jugar a la consola y crear software para ella. Para obtener información más detallada al respecto, se recomienda consultar los textos de licencia incluidos en cada uno de los programas disponibles.

## Sobre este documento

Este documento se proporciona bajo la Licencia de Atribución Creative Commons 4.0 (CC BY 4.0). Puede leerse el texto completo de la licencia en el sitio web de Creative Commons: <https://creativecommons.org/licenses/by/4.0/>

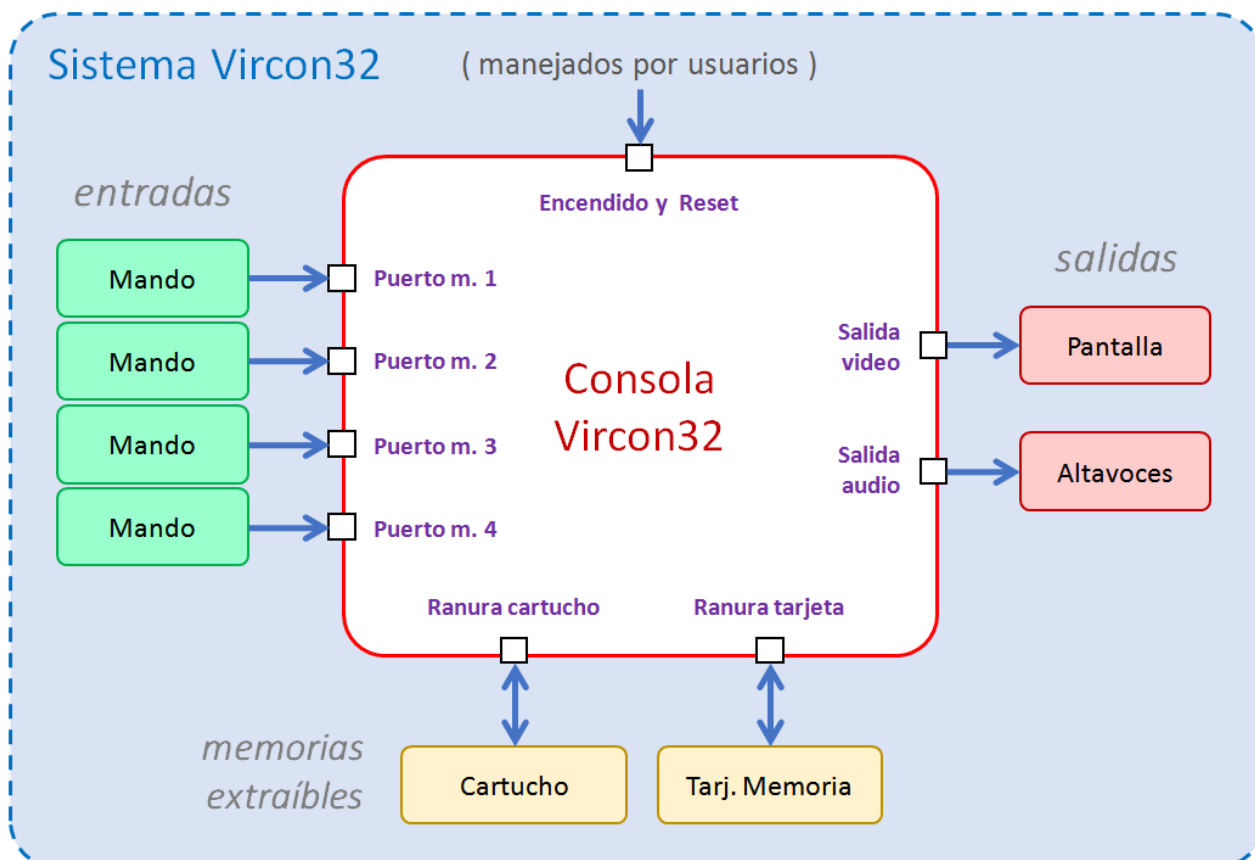
# Índice

Los anteriores documentos de la especificación ya definieron todos los elementos dentro de la consola. Ahora necesitamos describir los componentes que, aún formando parte del sistema Vircon32, son externos a la propia consola. La parte 1 de la especificación ya daba una visión básica de estos elementos externos, y ahora la parte 8 proporciona una descripción completa.

1 El sistema completo	3
2 Mandos	4
3 Pantalla	7
4 Altavoces	8
5 Cartuchos	10
6 Tarjetas de memoria	12

# 1 El sistema completo

Un sistema Vircon32 no incluye sólo la consola en sí: también se necesitan otros componentes externos a la consola para que funcione. Un sistema Vircon32 completo consistirá entonces en la consola junto con todos los posibles elementos externos conectados a ella. Podemos ver esos elementos y sus conexiones en este diagrama:



La consola es el elemento central de este sistema, y todos los elementos externos están concebidos para interactuar exclusivamente con ella. Para todos ellos, las siguientes secciones detallan sus funciones, su comportamiento, y su comunicación con la consola.

## 1.1 Elementos externos extraíbles

En general, se considera que todos los dispositivos externos son extraíbles: es concebible (aunque no muy útil) que la consola se encienda sin ninguno de ellos presente. También podría haber cualquier combinación de dispositivos presentes y ausentes.

Esto significa que, para ser totalmente conformes, las implementaciones deben soportar la opción de conectar y desconectar cada dispositivo externo. Para las implementaciones por software, esto debería ser trivial en la mayoría de los casos. Sin embargo, en algunos tipos de hardware puede que no sea posible.

Aunque hasta ahora el hardware de Vircon32 se ha representado como una consola doméstica, podría haber versiones hardware en otros formatos, como máquinas arcade o portátiles. En éstos puede ser aceptable para una implementación que la pantalla, los altavoces y/o los mandos sean fijos y no puedan separarse o desconectarse. Un ejemplo de ello podría ser este concepto de mueble arcade Vircon32:



En un sistema como este, la pantalla, los altavoces y los 4 mandos estarían integrados en la carcasa. Eso significa que estarán siempre presentes y permanecerán conectados. Esto es aceptable siempre que el sistema soporte todos los elementos de un sistema completo. Si, por el contrario, este mueble arcade fuera más delgado y sólo incluyera 2 joysticks, no se consideraría totalmente conforme a menos que se pudieran conectar 2 mandos adicionales de alguna forma.

En cuanto a los componentes de almacenamiento extraíble (cartuchos y tarjetas de memoria), se requiere que sean extraíbles en cualquier implementación porque están pensados para ser intercambiables. Esto también puede verse en las imágenes, donde ambos elementos se insertan a través de ranuras.

## 2 Mandos

Los mandos permiten a los jugadores interactuar con el juego pulsando direcciones y botones. La consola leerá su estado cada frame, y los juegos usarán esta información para realizar acciones o tomar decisiones. Se pueden conectar hasta 4 mandos a la consola.

Los mandos de Vircon32 se consideran todos idénticos en capacidades y diseño. La consola no puede distinguirlos, aparte del puerto concreto al que se conecta cada uno.

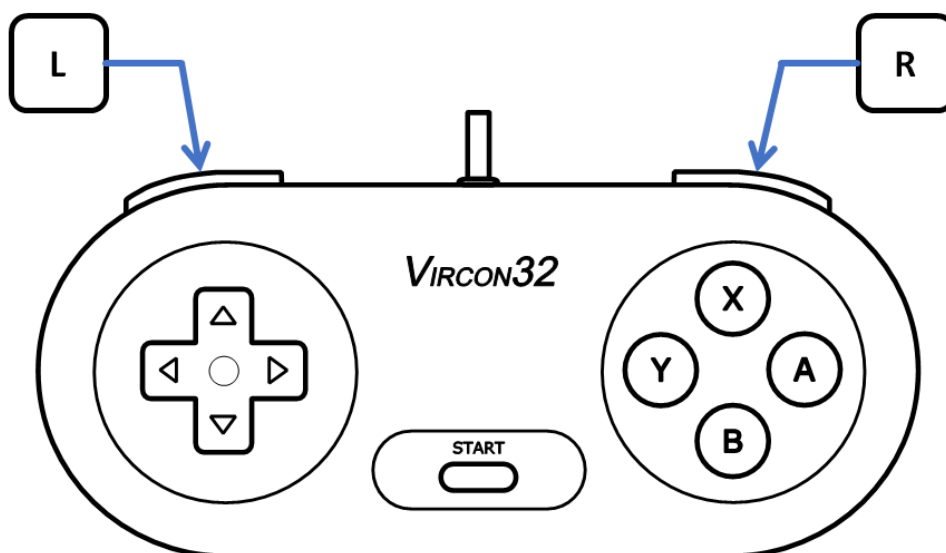
## 2.1 Controles y su colocación

Cada mando cuenta con 11 controles digitales. Estos controles se dividen en 2 categorías (4 direcciones y 7 botones) como se muestra en estas listas:

<b>Direcciones:</b> Izquierda, Derecha, Arriba, Abajo	<b>Botones:</b> A, B, X, Y, L, R, Start
--	--

Cada control sólo tiene 2 estados posibles, y los mandos los representan con 1 bit cada uno, siendo 1 = pulsado y 0 = no pulsado. Para todos los controles, su estado neutro es sin pulsar: cada control sólo se pulsa cuando el usuario realiza activamente alguna acción, como aplicar una fuerza.

El mando de Vircon32 organiza físicamente sus controles en grupos separados: una cruceta (Izquierda, Derecha, Arriba, Abajo), 4 botones frontales (A, B, X, Y), 2 botones superiores (L, R) y un botón central (Start). Para todas las implementaciones, los controles de cada mando deben seguir la colocación que se muestra en esta imagen.



Esto significa que los creadores de juegos pueden asumir con seguridad esta colocación de los controles al decidir su sistema de control. Por ejemplo, un juego de disparos podría asumir la forma de diamante de los 4 botones frontales para hacerlos funcionar como una "cruceta secundaria" que controle en qué dirección disparar (X dispara hacia arriba, Y dispara a la izquierda, etc.).

Los mandos de Vircon32 no incluyen ningún mecanismo adicional o automatización para los controles del mando. Esto significa que no tienen modos turbo, auto-disparo, auto-retención, ni son programables de ninguna manera.

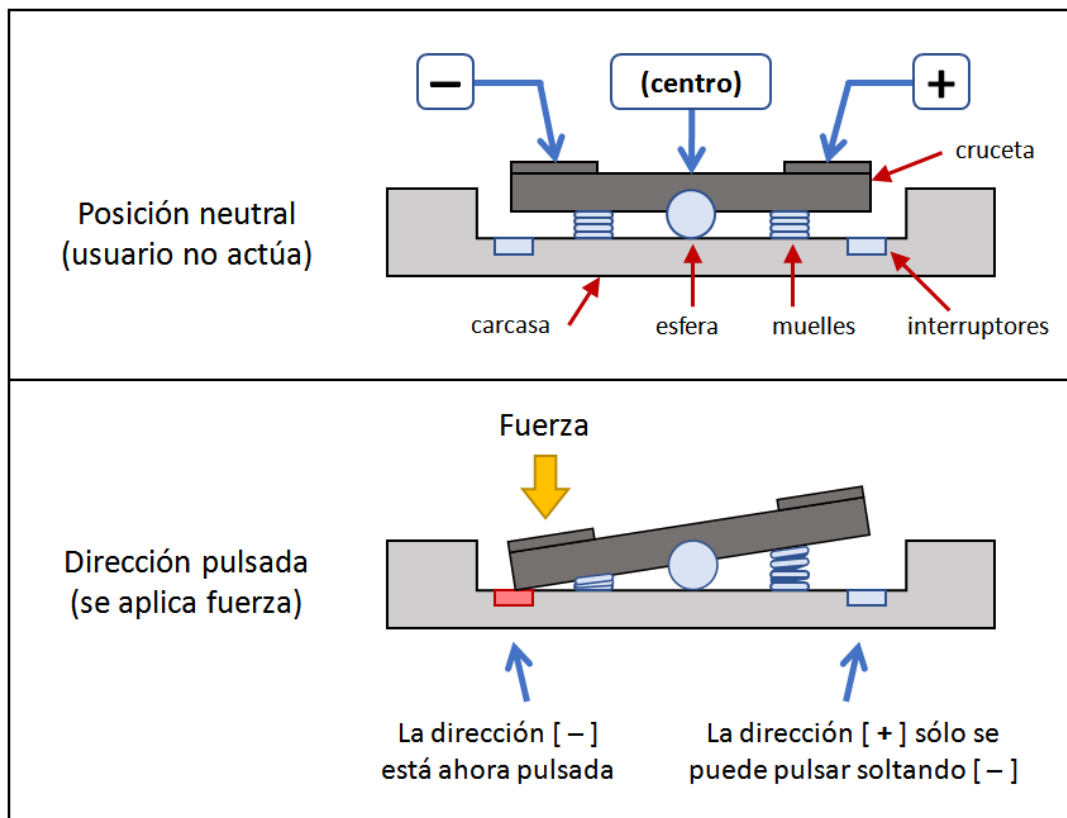
## 2.2 Cruceta

La cruceta puede implementarse de diferentes maneras, a elegir por cada aplicación. Las opciones posibles son: 4 botones separados, una forma de cruz, un octógono para tener diagonales explícitas, etc. Para cualquier opción elegida, su salida debe ser siempre de 4 bits (uno por cada una de las 4 direcciones).

En Vircon32 todas las crucetas deben implementar algún mecanismo para evitar que se puedan pulsar direcciones opuestas a la vez. Esto es necesario para asegurar que la consola siempre puede leer una dirección consistente en la cruceta. Esto significa que:

- El eje horizontal debe impedir que se pulsen a la vez Izquierda y Derecha.
- El eje vertical debe impedir que se pulsen a la vez Arriba y Abajo.

Esto se puede conseguir con diversas soluciones. Por ejemplo, muchas consolas clásicas usaban un mecanismo basculante. La siguiente imagen muestra una solución de este tipo con un diagrama sencillo. Si consideramos los dos extremos de un mismo eje, llamados aquí [ - ] y [ + ], tendríamos:



## 2.3 Conexión a la consola

La comunicación de los mandos con la consola se concibe como totalmente unidireccional: la consola no puede influir de ninguna manera en el estado o comportamiento de los mandos. La consola sólo leerá el estado de los mandos a través de cada uno de los puertos.

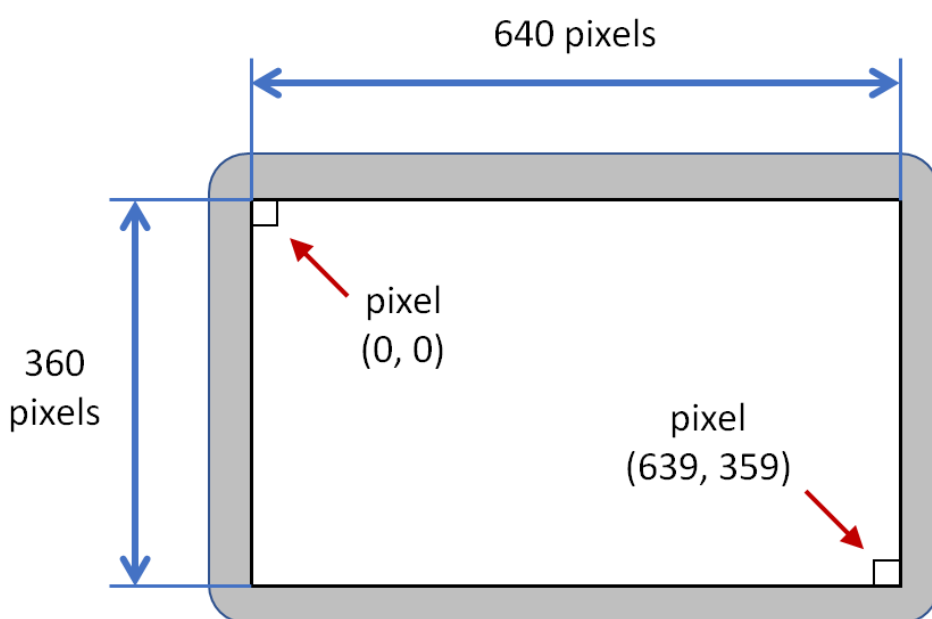
Las implementaciones pueden elegir cualquier cableado y conector, así como cualquier protocolo para las señales eléctricas y transmisión de datos. Sin embargo, los conectores elegidos deben incluir algún tipo de señal o sensor que permita a la consola saber en todo momento si un mando está conectado a cada uno de los 4 puertos. Esta información se representa como un único bit: 1 = conectado y 0 = desconectado.

## 3 Pantalla

La pantalla se usa para mostrar la imagen producida por la consola y hacerla visible para los jugadores. Recibirá la información del color de los pixels que le envía la GPU en cada frame, y la usará para producir la imagen pedida.

### 3.1 Dimensiones de la pantalla

La pantalla de Vircon32 es una matriz rectangular de 640 pixels en horizontal x 360 pixels en vertical (un total de 230400 pixels). Los pixels son cuadrados (relación de aspecto 1:1) y no tienen espacios entre ellos, por lo que la relación de aspecto para la pantalla completa resulta de 16:9. El tamaño de la pantalla es irrelevante para nuestros propósitos, y puede elegirlo libremente cada implementación.





Los pixels de la pantalla se numeran desde arriba a la izquierda (0,0) hasta abajo a la derecha (639,359). La pantalla no tiene distintos modos de vídeo o configuraciones: las propiedades de la imagen serán las mismas para todos los juegos e implementaciones.

Se asume que la pantalla de Vircon32 siempre se ve en una posición horizontal fija. Los creadores de juegos no pueden suponer que la pantalla pueda girarse de ninguna forma, ya que no se requiere que las implementaciones tengan esta capacidad. Esto significa, por ejemplo, que ningún juego de Vircon32 debería exigir a los usuarios girar la pantalla 90 grados, para tener una imagen vertical con relación de aspecto 9:16.

## 3.2 Calidad de la imagen

La pantalla muestra colores en el espacio RGB, con un rango de 24 bits (color verdadero). Para cada pixel hay 3 canales para rojo, verde y azul. Cada canal tiene una profundidad de color de 8 bits, de 0 (negro puro) a 255 (intensidad total). La pantalla produce vídeo sólo con el color recibido de cada pixel: la imagen final no tiene mezcla de color ni filtros.

La pantalla muestra 60 frames por segundo. Siempre produce video con frames completos: no hay interpolación ni entrelazado. Las pantallas reales siempre tendrán un cierto retardo de entrada, y las implementaciones deben decidir cómo gestionar esto.

## 3.3 Conexión a la consola

La comunicación de la consola con la pantalla se concibe como totalmente unidireccional: la pantalla no puede influir de ninguna manera en el comportamiento o el estado de la consola. La consola sólo enviará la señal de vídeo a través de su conector, y ni siquiera tiene forma de determinar si hay una pantalla conectada a ella.

Las implementaciones pueden elegir cualquier cableado y conector, así como cualquier protocolo para las señales eléctricas y la transmisión de datos. Sin embargo, la comunicación debe poder garantizar que cada frame se muestre en el momento correcto.

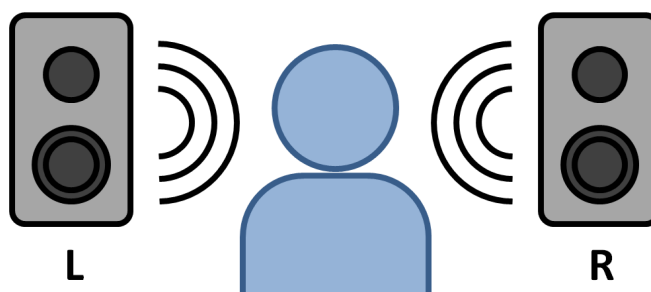
Como se indica en los documentos de la GPU y la SPU, en la práctica es habitual que una pantalla integre también funciones de audio. Los protocolos de comunicación actuales, como HDMI, pueden unir audio y vídeo en la misma señal y, por tanto, compartir cables y conectores. Por ello, aunque la pantalla y los altavoces se tratan aquí como entidades separadas, sería válido que una implementación uniera las salidas de audio y vídeo en una señal combinada enviada a un único dispositivo.

# 4 Altavoces

Los altavoces se utilizan para reproducir el sonido generado por la consola y hacer que los jugadores puedan oírlo. Recibirán la secuencia continua de samples que les envíe la SPU, y la usarán para producir el sonido pedido.

## 4.1 Canales de sonido

Vircon32 utiliza un conjunto estándar de altavoces 2.0 para producir sonido estéreo. Esto significa que hay 2 fuentes de sonido independientes con diferente colocación respecto a los jugadores: una a su izquierda (L) y otra a su derecha (R).



Este documento trata cada fuente de sonido como un altavoz independiente, pero puede haber implementaciones en las que un único dispositivo integre 2 fuentes independientes para izquierda y derecha.

## 4.2 Calidad del sonido

El sonido se produce a una frecuencia de muestreo de 44100 Hz, y los samples tienen una precisión de 16 bits. Cada altavoz recibirá una secuencia constante de samples de la SPU, y la usará para producir el sonido necesario para ese canal (izquierdo o derecho).

Se pueden usar auriculares como altavoces, pero siempre debe existir una fuente de sonido que puedan escuchar los 4 jugadores. No debe haber diferencias de calidad, retardos o distorsiones en el sonido escuchado por ninguno de los jugadores.

Los altavoces de Vircon32 producen sonido sólo con los valores recibidos para cada sample de la secuencia: el sonido final no tiene filtros ni ajustes de ecualización.

## 4.3 Conexión a la consola

La comunicación de la consola con los altavoces se concibe como totalmente unidireccional: los altavoces no pueden influir de ninguna manera en el comportamiento o el estado de la consola. La consola sólo enviará la señal de audio a través de su conector, y ni siquiera tiene forma de determinar si hay un altavoces conectados a ella.

Las implementaciones pueden elegir cualquier cableado y conector, así como cualquier protocolo para las señales eléctricas y la transmisión de datos. Sin embargo, la comunicación debe poder separar las secuencias de los altavoces izquierdo y derecho, y garantizar que cada sample se escucha en el momento correcto.

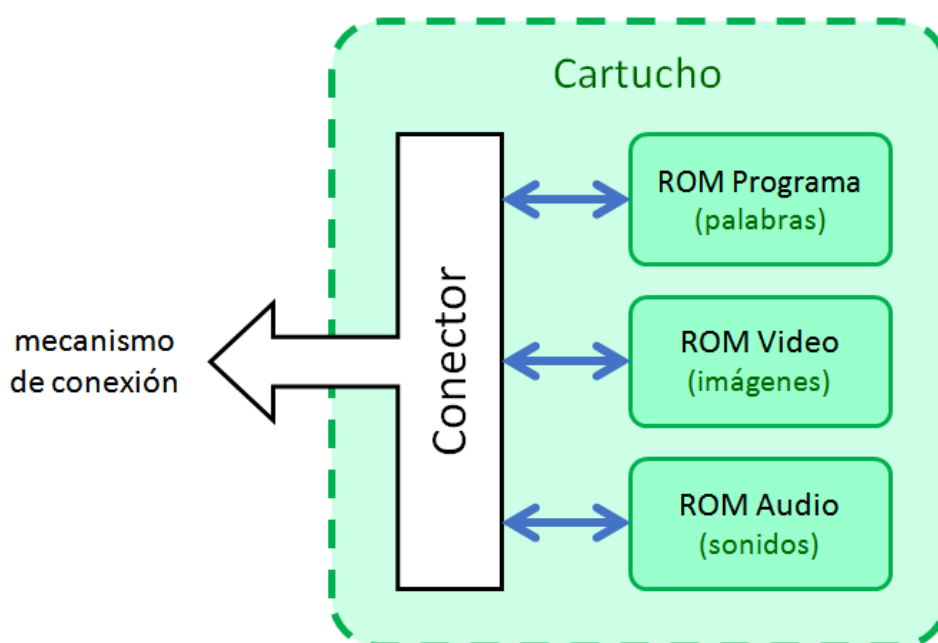
Como ya se mencionó para la pantalla, aunque la pantalla y los altavoces se tratan como entidades separadas, sería válido que una implementación uniera las salidas de audio y vídeo en una señal combinada enviada a un único dispositivo.

## 5 Cartuchos

Los cartuchos son el principal soporte de almacenamiento en Vircon32. En general, cada cartucho almacenará un juego separado y la consola ejecutará su programa desde el cartucho. Cada cartucho puede tener un contenido diferente, pero todos tienen la misma estructura y conectores para hacerlos intercambiables.

### 5.1 Estructura de un cartucho

Los cartuchos de Vircon32 pueden contener hasta 3 memorias de sólo lectura (ROM) independientes: programa, vídeo y audio. La comunicación desde la consola se llevará a cabo a través de algún tipo de conector que proveerá información sobre todas las ROMs del cartucho y gestionará el acceso a su contenido.



Todos los cartuchos deben contener una ROM de programa, pero las ROMs de audio y vídeo son opcionales. Cada una de las 3 ROMs, cuando está presente, puede tener un número variable de elementos (palabras / imágenes / sonidos) en diferentes cartuchos.

La consola debe tener alguna forma (elegida por la implementación) de determinar el contenido de cada cartucho. Esto puede hacerse incluyendo información adicional en los cartuchos (cabeceras o metadatos), o diseñando la comunicación consola-cartucho para que permita a la consola conocer la estructura interna de cada cartucho en particular.

Las 3 ROMs almacenan sus contenidos sin ninguna compresión, y para cada una el límite de capacidad de almacenamiento es el siguiente:

- ROM de Programa: hasta 128 MWords =  $128 \times 1024 \times 1024$  palabras = 512 MB.
- ROM de Video: hasta 256 texturas =  $256 \times 1024 \times 1024$  pixels = 1 GB.
- ROM de Audio: hasta 1024 sonidos, y  $256 \times 1024 \times 1024$  samples = 1 GB.

Esto hace que la máxima capacidad posible para un cartucho sea en total de 2.5 GB.

## 5.2 Rendimiento de la memoria

Aunque no se indican niveles de rendimiento, se asume que las 3 ROMs de cartucho son lo bastante rápidas como para ser capaces de lo siguiente:

- En cada ciclo de CPU, la ROM de programa puede enviar a la CPU todas las palabras que pide a tiempo para que termine la instrucción en ese mismo ciclo.
- En cada frame, la ROM de vídeo puede enviar a la GPU todos los pixels que pide a tiempo para terminar todas las operaciones de dibujo en ese mismo frame.
- En cada frame, la ROM de audio puede enviar a la SPU todos los samples que pide a tiempo para terminar la generación de sonido en ese mismo frame.

## 5.3 Conexión a la consola

La comunicación de la consola con los cartuchos se concibe como totalmente unidireccional: los cartuchos son componentes pasivos y, como toda su memoria es de sólo lectura, la consola sólo puede leer su contenido, pero no influir en su estado de ninguna forma.

Las implementaciones pueden elegir cualquier cable y conector, así como cualquier protocolo para las señales eléctricas y la transmisión de datos. Sin embargo, hay algunos requisitos que deben cumplir los conectores y/o protocolos elegidos:

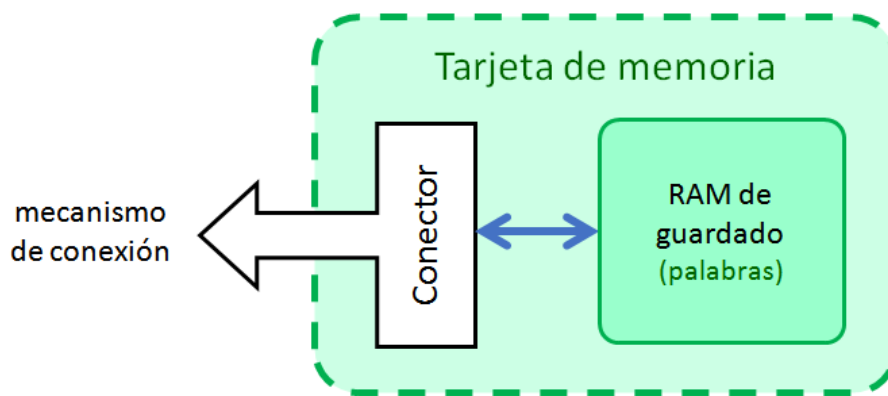
- Deben incluir algún tipo de señal o sensor que permita a la consola saber en todo momento si hay un cartucho conectado a su ranura. Esta información se representa como un único bit: 1 = conectado y 0 = desconectado.
- La comunicación elegida debe ser la misma en todos los cartuchos diseñados para esa implementación concreta, para garantizar que todos ellos sean intercambiables.

## 6 Tarjetas de memoria

Las tarjetas de memoria dotan a la consola de un pequeño almacenamiento permanente, que los juegos pueden usar para cargar y guardar datos entre distintas sesiones. A diferencia de los cartuchos, todas las tarjetas de memoria Vircon32 se consideran idénticas. La consola sólo puede distinguir unas de otras examinando su contenido.

### 6.1 Estructura de una tarjeta de memoria

Todas las tarjetas de memoria de Vircon32 son funcionalmente idénticas y, como tales, comparten la misma estructura y capacidad de almacenamiento. Las tarjetas disponen de una memoria interna de acceso aleatorio (RAM) que permite leer y escribir de forma separada cualquiera de las palabras que contiene. La comunicación con la consola se hará a través de algún tipo de conector que gestione el acceso a la memoria interna.



A diferencia de la RAM de la consola, la RAM de la tarjeta de memoria almacena su contenido de forma permanente. No se borra al apagar/reiniciar, o cuando se extrae la tarjeta de la consola. Las tarjetas de Vircon32 guardan su contenido sin ninguna compresión, y todas tienen una capacidad de 256 KWords = 256 x 1024 palabras = 1MB.

### 6.2 Rendimiento de la memoria

Aunque no se indican niveles de rendimiento, se asume que la memoria de la tarjeta es lo bastante rápida como para responder a todas las peticiones de la CPU a tiempo para terminar cada instrucción dentro del mismo ciclo.

### 6.3 Conexión a la consola

La comunicación de las tarjetas de memoria con la consola es bidireccional. Aunque una tarjeta de memoria es un componente pasivo (la comunicación siempre se inicia desde la consola), la memoria de la tarjeta se puede leer y escribir. Por tanto, a diferencia de los cartuchos, la consola puede modificar el contenido de una tarjeta de memoria.

Las implementaciones pueden elegir cualquier tipo de cable y conectores, así como cualquier protocolo para las señales eléctricas y la transmisión de datos. Sin embargo, los conectores elegidos deben incluir algún tipo de señal o sensor que permita a la consola saber en todo momento si hay una tarjeta de memoria conectada a su ranura. Esta información se representa como un único bit: 1 = conectada y 0 = desconectada. Además, la comunicación elegida tiene que ser la misma para todas las tarjetas diseñadas para esa implementación concreta, para garantizar que todas ellas sean intercambiables.

*( Fin de la parte 8 )*