

## COMO FABRICAR UN CARTUCHO PROGRAMABLE

Por David Senabre. 2004.

### INTRODUCCION

Aquí explicaré como modificar un cartucho de SNES para poder usar memorias ROM programables, y por lo tanto poder meter vuestros propios programas, aunque la mayoría de vosotros estaréis interesados en meter juegos, obviamente. Eso sí, no entréis en éxtasis, porque únicamente podremos usar esta cartucho para juegos de 512Kb y 1Mb, sin posibilidad de salvar partida. Pero es un proyecto interesante y se aprende mucho de cara a hacer algo más complicado. No creo que muchos escribáis programas para la SNES, pero si lo hacéis, esta es una buena manera de probarlos en una SNES real. Por cierto, para aquellos que tengan un mínimo interés y ni puta idea, programar para la SNES no es muy difícil, así que ya sabes, aprende.

### ¿EN QUE SE BASA?

La modificación se basa en sustituir la MROM (Mask ROM) que trae el cartucho, con una memoria EEPROM, o Flash ROM, en mi caso. Ambas son regrabables, mientras que la que trae el cartucho no lo es. Pero Nintendo usó unas MROM con un pinout no estándar, lo que significa que no corresponde con ningun pinout de las memorias EEPROM/Flash actuales, supongo que para evitar que la gente hiciera justamente lo que estoy explicando :) Otros dicen que los distintos pinouts eran para optimizar el diseño de las placas, pero no me parece muy probable. Nintendo siempre tiene que utilizar cosas que no use nadie más. Siempre ha sido su filosofía.

Las diferencias de los pinouts radican en un par de líneas de dirección, y la señal /RD. Por ello debemos redirigir las líneas que difieren, al pin correspondiente de la nueva ROM que vayamos a usar. Es sencillo, y comprendiendo como funciona esto, es sólo cuestión de pensar.

### ACLARACIONES

Yo voy a hacer un ejemplo para un cartucho en particular, pero los mismos razonamientos se aplican a cualquier otro cartucho (en principio, así que no vayáis a coger un cartucho que traiga SuperFX o algún chip especial, primero porque si es vuestra primera vez y/o no sabéis bien lo que hacéis, podéis cargároslo, y esos cartuchos son más valiosos, y segundo porque los integrados de esos cartuchos son de otro formato distinto, más difícil de manipular si no se tiene experiencia)

Así que encuentra un cartucho normal. Yo mostraré el ejemplo con la placa SVCH-2A0N, LoROM, de un cartucho Street Fighter 2. Por lo tanto este juego sólo funcionará con ROMs LoROM.

## EJEMPLO PARA EL SHVC-2A0N

### INTRODUCCION

Este cartucho mola especialmente porque viene con 2 MROMs de 32 pines, luego podremos meter el doble de capacidad que si viniera con una sola MROM, que es lo habitual. Para quien no entienda porqué, esto es simplemente porque podemos soldar 2 EEPROMs en lugar de una. No vuelvo a hacer una aclaración tan chorra, aviso.

### EL DECODIFICADOR

Para empezar, debemos conocer si el cartucho trae decodificador de direcciones o no. Si el cartucho tiene más de una MROM y/o una SRAM, siempre traerá un decodificador. Éste se encarga de decidir a quién va

dirigida una determinada dirección. En mi ejemplo tenemos el 74LS00, que no es más que un integrado con 4 puertas NAND, y su pinout es el que sigue.

1A		01\		14		+5V
1B		02		13		4B
1Y		03		12		4A
2A		04		11		4Y
2B		05		10		3B
2Y		06		09		3A
GND		07		08		3Y

Los números se refieren a la puerta NAND en cuestión. La A y B son las entradas a cada NAND, e Y es la salida de la NAND. En el ejemplo de ahora, este integrado está conectado al cartucho así:

@1		01\		14		+5V
A20		02		13		A20
ROM2 /RD		03		12		+5V
@2		04		11		@2
@1		05		10		#49
ROM1 /RD		06		09		+5V
GND		07		08		@1

El simbolo '@' significa que los pines que lo tienen están conectados. #49 significa pin 49 del conector de cartuchos, que es /ROM RD

Si analizas el funcionamiento de este integrado, te darás cuenta de que lo único que hace es activar la ROM 1 cuando la línea A20 está a 0, y activar la ROM 2 cuando la línea A20 está a 1.

ROM 1 : 000000 - 0FFFFFFF  
 ROM 2 : 100000 - 1FFFFFFF

Pues ya está, esto quiere decir que para la mitad de las direcciones accederá a una ROM, y para la otra mitad, a la otra. En este caso las MRoms son de igual capacidad, de 1Mb cada una, pero nosotros las vamos a sustituir por 2 Flash ROM 29F040, que tienen la mitad de capacidad, por lo que habrá que modificar el decodificador para que pueda acceder correctamente a la segunda ROM.

¿Cómo? Como nuestro cartucho tendrá en total 1Mb de capacidad, dividido en 2 EEPROMs de 512Kb, serán necesarias 20 líneas de dirección para acceder a ellas (2 elevado a 20 = 1Mb). Recordemos que en los diagramas y pinouts, normalmente se empieza a contar desde A0, A1, A2, etc... luego A0 es la primera, A1 es la segunda,... Cuidado. Por lo tanto la línea número 20 es la A19. Esas son las líneas que necesitamos para acceder a nuestras ROMs, desde A0 a A19, pero la A20 ya no, porque hemos perdido la mitad de capacidad respecto al diseño original.

La mitad de las direcciones tendrán A19 a 0, y la otra mitad a 1, por lo que usaremos esta línea en lugar de A20 en nuestro decodificador, para activar una u otra ROM. Facil, ¿no?

ROM 1 : 00000 - 7FFFFF  
 ROM 2 : 80000 - FFFFFF

El porqué no usamos una EEPROM/Flash ROM de 1Mb de capacidad o más, es porque las Fash ROM que yo uso, por encima de los 512 Kb ya no está

disponible en formato DIP, y personalmente no me gustan las EEPROM de la serie 27Cxxx, además de que no tengo programador para ellas :)

### SUSTITUIR LAS MASK ROMs

Existen 2 métodos diferentes:

Método conservativo : Más difícil. Se trata de desoldar las pastillas MROM de manera que queden intactas. Se hace con un chupador de estaño y malla desoldadora, y paciencia. Debes quitar todo el estaño de todas las patas antes de que la MROM pueda salir. Es un coñazo, y sólo lo recomiendo si deseas por cualquier motivo que la MROM no se joda. Pero aún así debes tener cuidado porque si calientas demasiado la MROM, puedes cargártela igual.

Método Destructivo : Rápido y sencillo. Se trata de cortar las patas de la MROM por la parte superior, y luego desoldar las patas, que han quedado sueltas, una a una, sujetando un extremo por un lado con unos alicates, y aplicando el soldador por la otra cara. Para mí es un método sencillo, porque uso una Dremel para cortar las patas. Eso sí, mucho cuidado de que no se os vaya y cortéis alguna pista. Os lo digo por experiencia... así que cuidado. No hace falta una dremel, unos buenos alicates de corte sirven.

Sobra decir que la MROM no volverá a funcionar, a no ser que seas el puto amo del soldador y vuelvas a soldar las patas a la MROM. Si lo intentas verás que no es sencillo. Para hacer este cartucho, yo usé el método destructivo-Dremel :)

### LOS PINOUT DE LAS MROM y Flash ROM

El pinout de la MROM que usaba Nintendo es:

(los paréntesis indican los pines que se diferencian de la ROM que usaré como sustituta, la Flash ROM 29F040).

A20		— \ / —		+5V
A21		— —		A22
(A18) A17		01 \ / 32		+5V
(A16) A18		02 31		/RD (/WR)
A15		03 30		A19 (A17)
A12		04 29		A14
A7		05 28		A13
A6		06 27		A8
A5		07 26		A9
A4		08 25		A11
A3		09 24		A16 (/RD)
A2		10 23		A10
A1		11 22		/CS
A0		12 21		D7
D0		13 20		D6
D1		14 19		D5
D2		15 18		D4
GND		16 17		D3

-----

### RECABLEADO

Se supone que ya hemos desoldado la MROM.

Como deberás haberte percatado, es necesario hacer un recableado de las pistas para compensar las diferencias entre ambas ROMs. A los pines que difieren entre una y otra los llamaré "pines diferentes".

También existen 2 métodos.

Reversible : Se usan dos zócalos. Al primero se le quitan los pines diferentes, y el resto se suelda al PCB del cartucho. Donde no hay pin, se suelda un cable. Sobre el primer zócalo se pone un segundo zócalo, sin necesidad de arrancarle ningún pin. Entonces se trata de soldar los cables, que están soldados a los pines diferentes del PCB, al pin correspondiente del zócalo. De este modo hemos redireccionado las líneas a su lugar correcto. No daré más detalles porque es un método bastante mecánico.

No reversible : Lo de no reversible no es definitivo, más bien sería "difícilmente reversible". O vamos, que sería un coñazo. Aquí sólo usaremos un zócalo con todos sus pines. Se trata de cortar las pistas que van a los pines diferentes del zócalo, y luego con trozos de cable, llevar las líneas correctas a esos pines que ahora flotan (eso quiere decir que están desconectados, ya que hemos cortado la pista a la que estaban conectados).



*Pista cortada. Puedes usar un cutter para ello. Comprueba con un polímetro cuando deja de haber continuidad.*

Este método tiene la ventaja de que ocupa menos espacio, y la desventaja de que los cambios son difíciles de rectificar porque implica cortar pistas del PCB. Es un método menos mecánico. Debes invertir tiempo en examinar las distintas opciones de cableado, donde cortar, y de donde traer las líneas correctas. Yo elegí este método.

## **RECABLEADO NO REVERSIBLE**

### **RECABLEADO DE LA ROM**

#### Aprovecha las bias

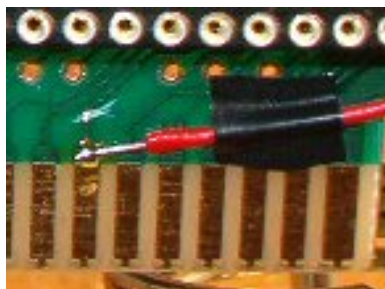
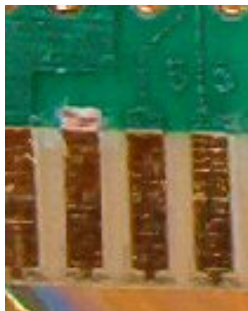
Las bias son esos agujeritos que tienen todos los PCB, y que se usan para llevar una pista de una cara a otra de la placa. En la placa de mi ejemplo, tanto **A17** y **A18** pasan por una bia. ¿Cómo aprovechar esto? Muy sencillo. Partimos del conector del cartucho.

Seguimos la pista hasta llegar a la bia.

Cortamos la pista \*después\* de pasar la bia.

Ya hemos dejado ese pin flotando. Ahora soldamos un cable a la bia, y lo llevamos al pin correcto. ¿Qué ventajas tiene esto? Soldar en una bia es muy sencillo. Basta con estañarlas un poco, y calentando el cable, introducirlo en la bia (es muy sencillo en el caso de estas placas, claro, porque en las placas modernas es otra historia, ya que las bias son diminutas y muy puñeteras).

**A16** no pasa por ninguna bia en este ejemplo. Bueno, nos hace un poco la puñeta pero no es el fin. El único método que se me ocurre es raspar la parte recubierta de verde del conector, para dejar que se vea el cobre que está debajo, estañarlo un poco, y soldar un extremo del cable ahí. Este cable debe ser largo, porque tendremos que llevarlo a la otra cara de la placa para soldarlo en su posición.



*Raspa como en este ejemplo*

Respecto a **/WR**, simplemente olvídate de ella. Corta la pista no sueldes ningún cable, o si lo prefieres, por seguridad, suelda un cable a +5V. Nunca a tierra ¿eh? **/WR** indica a una memoria que se está llevando a cabo una operación de escritura, y se activa cuando se pone a tierra. Si la conectas a tierra estará constantemente escribiendo y no funcionará. Lógicamente no queremos que esto ocurra. Mira que dije que no haría más avisos chorra. Si lo de **/WR** no lo sabías, es que necesitas saber un poco más antes de meterte en fregaos como este. Estas avisado. Otra opción es conectar **/WR** a SNES **#/WR**. Así se podría regrabar el cartucho sin sacar la Flash ROM del PCB, pero no entraré en detalles en este tutorial.

#### RECABLEADO DEL DECODIFICADOR

Para la ROM 1, ROM1 **/RD** pasa por una bia, así que lo tenemos fácil. Para la ROM 2, ROM2 **/RD** no pasa por ninguna bia. Con cuidado suelda el cable al pin correspondiente del decodificador, y de ahí al pin correspondiente del zócalo, como siempre. Como siempre desde hace unos minutos, pero bueno :)

Sólo queda sustituir A20 por A19 en el decodificador. El motivo de este cambio ya lo expliqué en el apartado de EL DECODIFICADOR. En este caso es muy fácil. A20 pasa por una bia, así que cortamos la pista, pero esta vez *\*antes\** de llegar a la bia desde el conector, ya que A20 no la necesitamos para nada, y soldamos en esa bia, que va derechita al decodificador, la línea A19.

Creo que con todo lo que he dicho deberías ser capaz de llevar a cabo esta modificación sin problemas. Sino, o no lo has entendido, o soy muy malo explicando :)

#### FOTOS DEL PROYECTO

Una imagen vale más que un tutorial soso, así que aquí os dejo las fotos del cartucho acabado para una Flash ROM. Funciona 100% :)  
SUERTE!!

