

VIDEOPAC FPGA

FPGA Core Odyssey2 / Videopac G7000

Edición del Charset

El nuevo Core que implementa el funcionamiento de la consola **Magnavox Odyssey2**, conocida en Europa como **Philips Videpac G7000** (en adelante Videopac) permite la personalización del juego de caracteres que reside en las máquinas originales en su chip gráfico y de sonido, el **Intel 8244/8245** <https://github.com/RW-FPGA-devel-Team/Videopac-G7000>

Para ello, desde su desde su menú OSD, en la opción "**Change VDC font *.CHR**" puedes cargar el juego de caracteres que más te guste.



Podrás encontrar algunos ya creados en la misma carpeta de este manual, como por ejemplo 'charset_videopac.chr' o 'charset_quads.chr'.

Se recomienda que tengas una carpeta llamada 'CHR' dentro de tu carpeta de juegos Videopac.

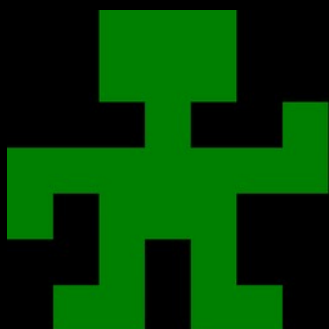
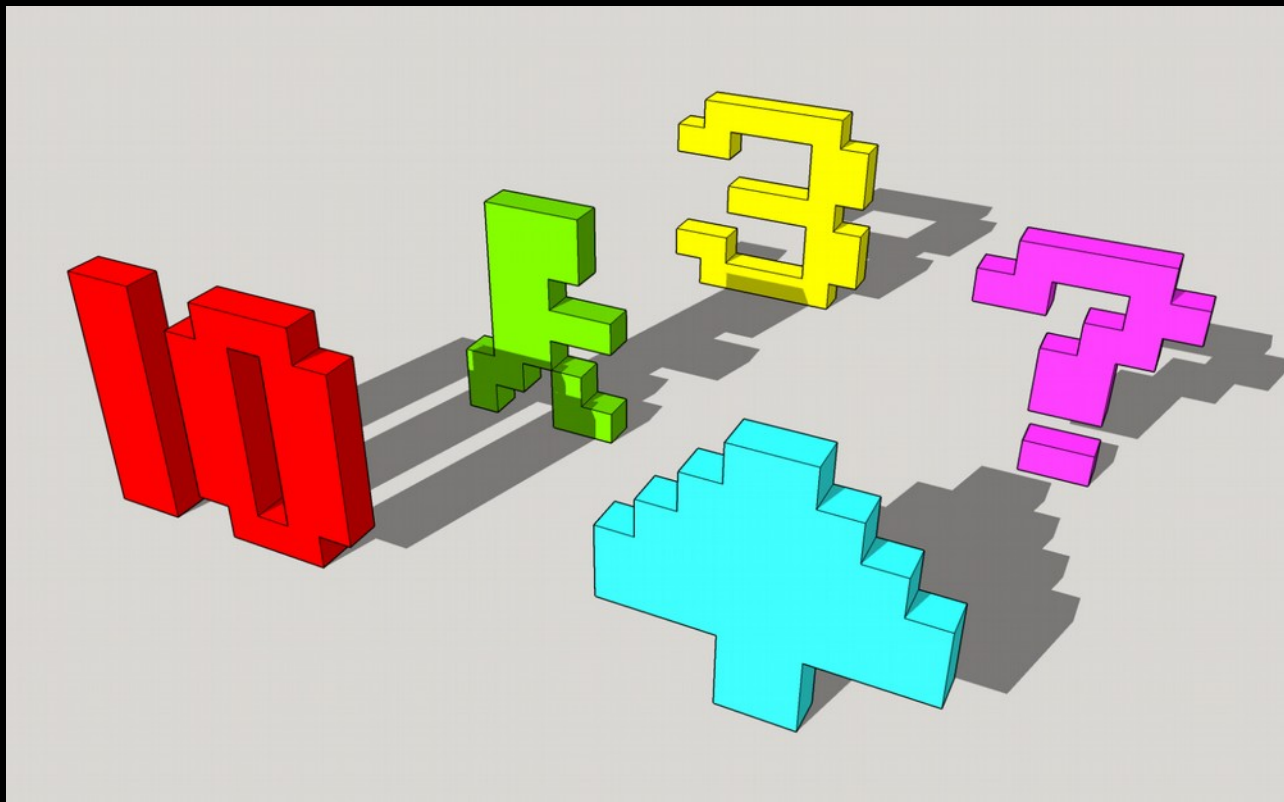
¿Cómo puedo modificarlos?

Su edición no puede ser más simple y para aprender no hay nada mejor que un ejemplo. Vamos a trabajar en un caso práctico:

El Videopac está capacitado para manejar sprites por hardware. Pero en ocasiones puede resultar divertido también (seas o no seas programador de videojuegos) la manipulación de los 64 caracteres que se encuentran en la ROM, concretamente dentro de su alma, el mencionado chip Intel custom VDC, que genera la señal de vídeo y los pulsos de audio, entre otras cosas.



Cada uno de esos 64 caracteres está formado por una celdilla de 8x8 pixels.



383812FEB8286C00

Para la manipulación de esos puntos de imagen disponemos de nuestro propio Charset Editor. Un pequeño código en html que te permite cómodamente apagar o encender cada pixel (botones izquierdo y derecho del ratón).

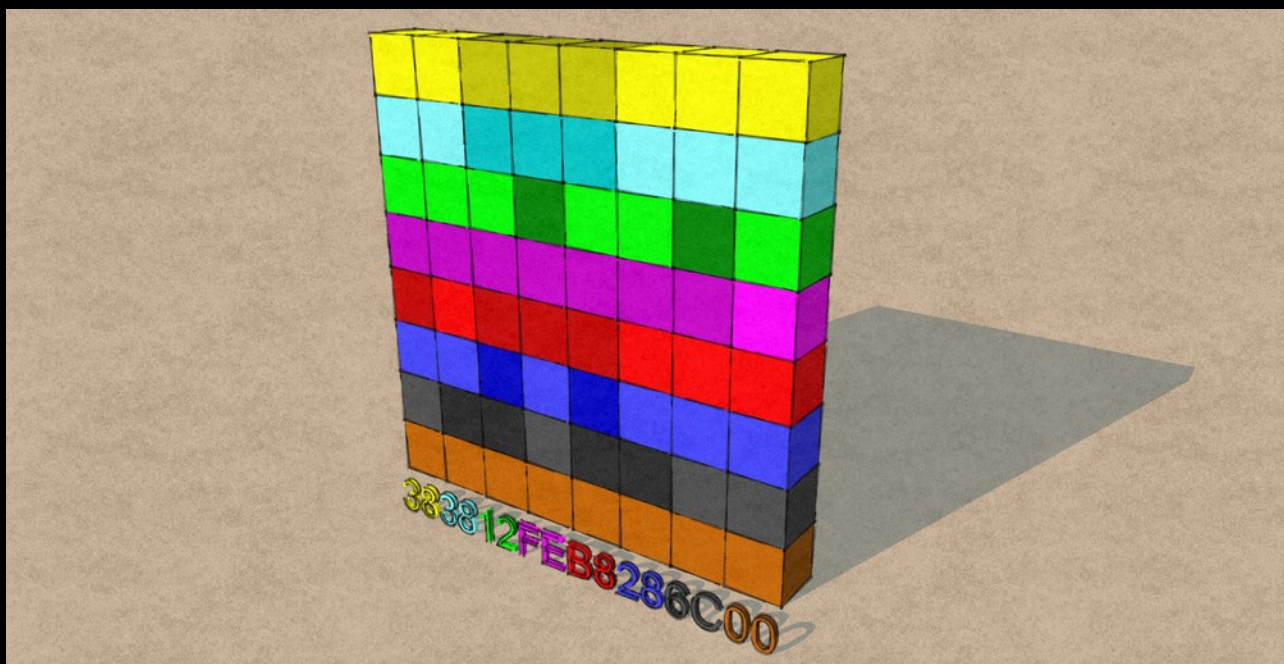


Podrás observar que en la parte inferior se muestra una cadena de números y letras. Se trata de los códigos hexadecimales que representan la imagen que estás creando.

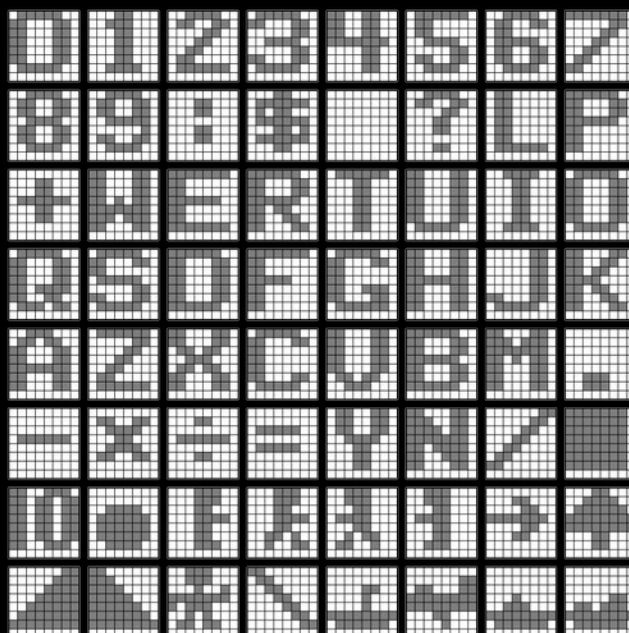
383812FEB8286C00

Cada línea de 1x8 pixels está almacenada ocupando un byte de memoria ROM en el chip. Por tanto cada carácter de 8x8 pixels consume 8 bytes de espacio.

Puedes ver en esta imagen los ceros y unos representados por tonos claros y oscuros, en 8 colores para los 8 bytes...



En el Videopac el espacio de memoria reservado para esos caracteres (64 en total) es de 512 bytes. Estos son los caracteres originales de la máquina. Con ellos vamos a trabajar...



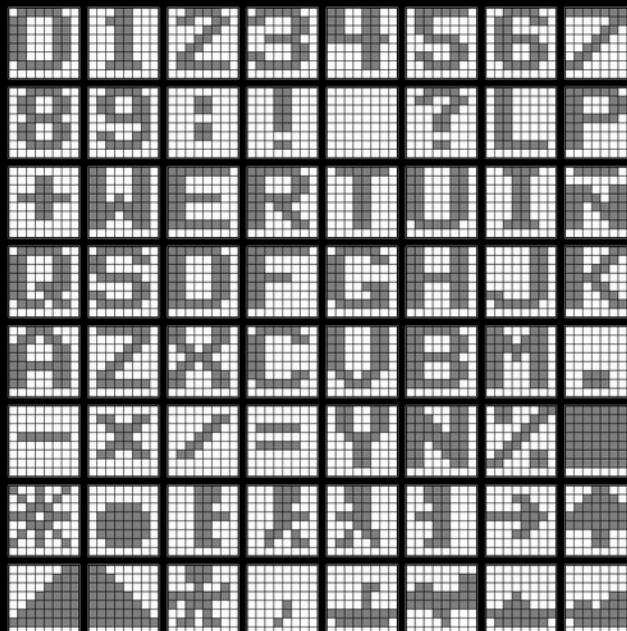
Algunas consideraciones

Como puedes ver, el número '0' y la letra 'O' tienen la misma configuración de pixels, por tanto podríamos prescindir de la letra 'O' y cuando la necesites, usa el cero en su lugar. Pero además podemos animarnos con otros cambios.

Tenemos otros caracteres que invitan a su manipulación:

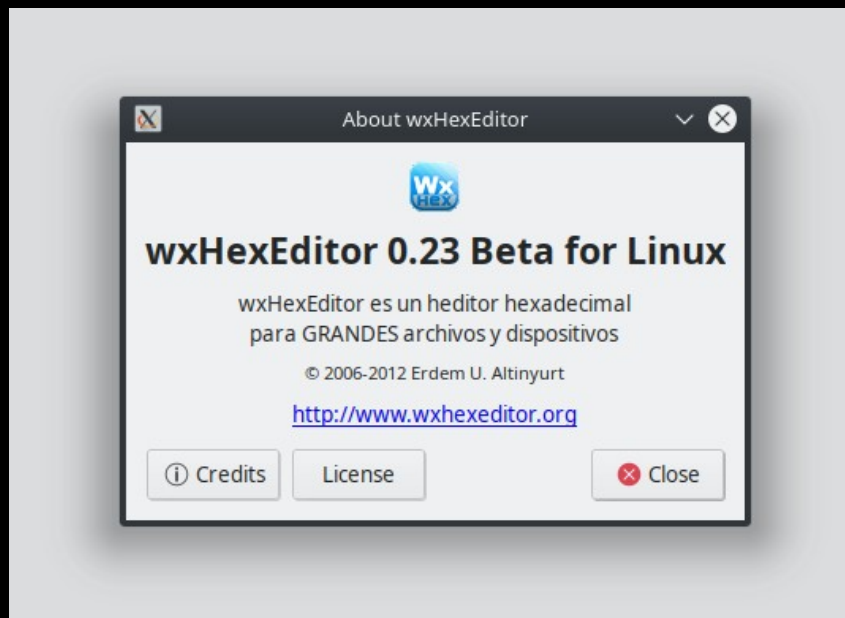
- En el carácter 'O' podríamos poner la letra 'ñ', tan necesitada en español.
- El carácter "\$" podría ser sustituido por "!". Pienso que en textos para videojuegos es más valorada la exclamación.
- El *signo división* puede ser mejorado con *barra inclinada* (Slash).
- El carácter de la *barra inclinada* original por el signo de *porcentaje*.
- El carácter "10" (que tiene poca utilidad práctica) por un *asterisco*.
- El carácter "," que lo pondríamos en el lugar de la *barra invertida* y nos daría doble servicio: La *tilde* para vocales acentuadas y la propia *coma*.

El *set* nos quedaría de esta forma...

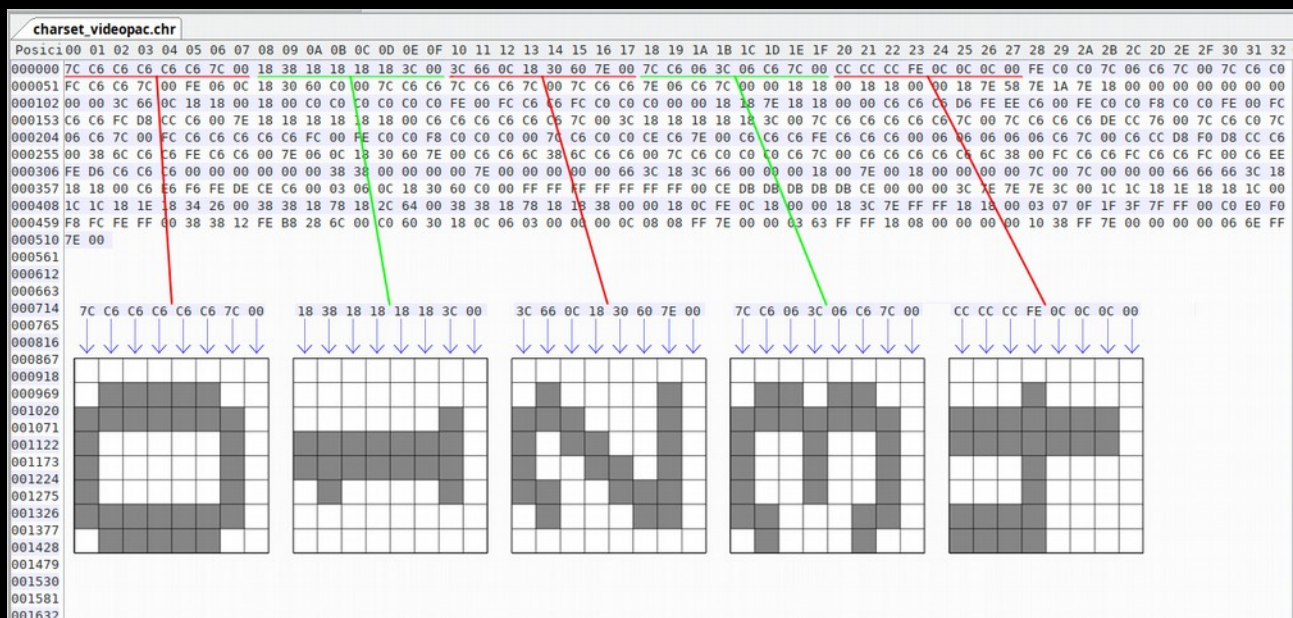


Vamos a necesitar 3 cosas:

- El archivo "*charset_videopac.chr*" de una longitud de 512 bytes
- El editor gráfico "*Charset_Editor.html*"
- Un editor hexadecimal cualquiera. En nuestro caso utilizaremos "*wxHexEditor*", pero puedes utilizar cualquier otro para *Linux*, *Windows* o *Mac*. Es indiferente.



La estructura de los datos es la siguiente: El primer byte (2 dígitos en hexadecimal) corresponde a la información de bits, ceros y unos, de la primera línea de puntos del gráfico, el segundo byte a la segunda, el tercer byte a la tercera, etc. Tras el octavo byte comenzará el primer byte del segundo carácter de la tabla, luego el segundo byte, tercero y siguientes, hasta completar los 64 caracteres de 8 bytes cada uno de toda la tabla.

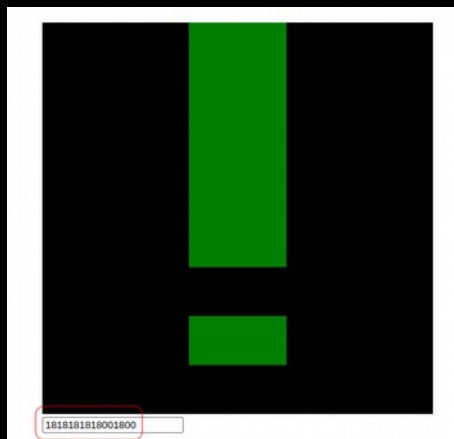


Bit a 0 indica que ese pixel en pantalla deberá tomar el color de fondo del juego y bit a 1 indica que ese pixel debe presentarse en pantalla con el color asignado por el programador para todo el carácter completo al que pertenece, es decir todos los 'unos' de sus 8 bytes, como se muestra en la imagen siguiente...

0	0	1	1	1	0	0	0
0	0	1	1	1	0	0	0
0	0	0	1	0	0	1	0
1	1	1	1	1	1	1	0
1	0	1	1	1	0	0	0
0	0	1	0	1	0	0	0
0	1	1	0	1	1	0	0
0	0	0	0	0	0	0	0

Vamos a empezar con el carácter "\$" que convertiremos en el carácter "!". Para ello, abrimos el archivo "*Charset_Editor.html*"

Usando el ratón podemos 'encender' y 'apagar' los pixels que necesitamos para formar la imagen de la exclamación. Observa que en los bytes que se muestran en hexadecimal, en la parte inferior, van cambiando en tiempo real. Todos los bytes, cuando tengas la imagen terminada tendrán el valor '00' o bien el valor '18', formando una cadena de 16 dígitos.

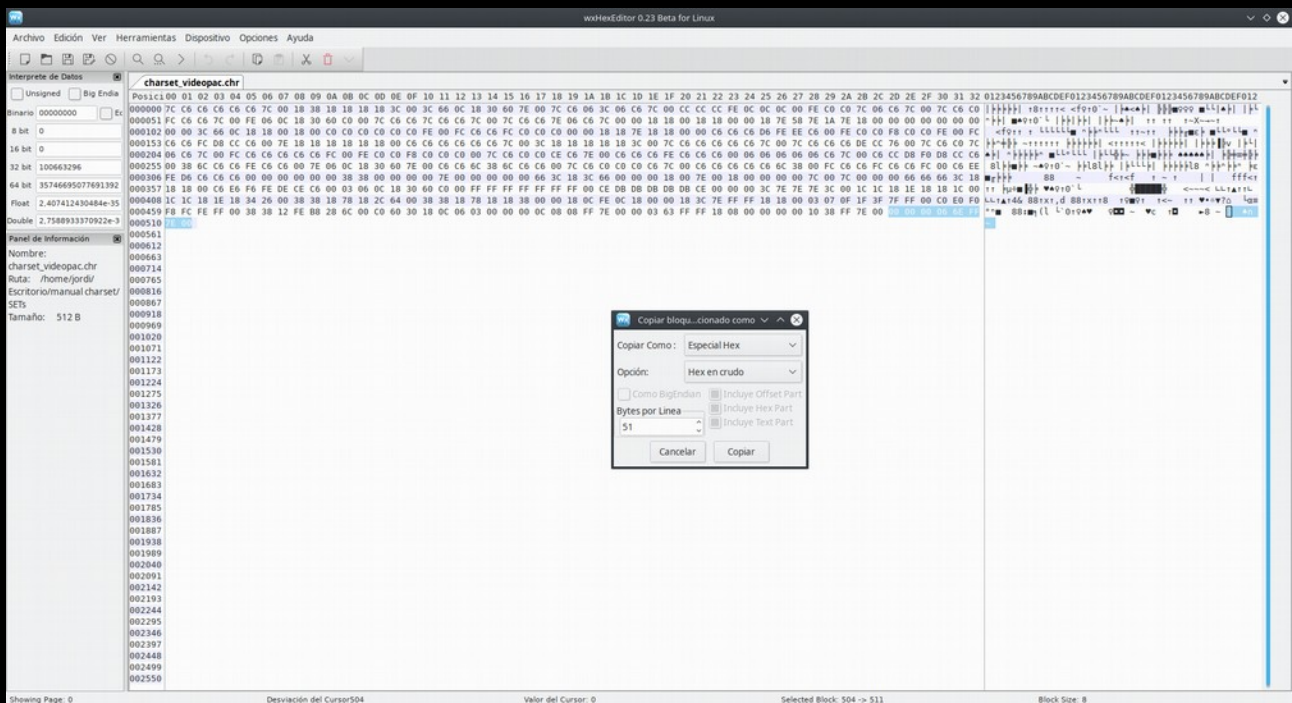


El editor también funciona a la inversa. Es decir que puedes escribir esos valores hexadecimales en ese campo de texto y al pulsar 'intro' se actualizará la matriz de puntos. Eso es especialmente útil si quieres buscar un carácter en concreto, desde el *wxHexEditor*, para poder editar sus 8 bytes con precisión.

Para ello simplemente debes seguir estos pasos:

Selecciona con el ratón un grupo de 8 bytes cualquiera, por ejemplo los 8 últimos bytes del archivo.

No necesitas contar los bytes que llevas seleccionados. En la Barra de Estado, el apartado "*Block Size:*" te indicará siempre cuantos tienes sombreados. Ahora desde el menú contextual eliges copiar como... y en el diálogo eliges "*Especial Hex*" y "*Hex en crudo*".



Pega ahora el contenido del portapapeles a la caja de texto hexadecimal del editor html y al pulsar 'intro' aparecerá esta imagen...



Como ya habrás imaginado, para colocar tu nuevo gráfico del signo de exclamación en el lugar correcto de la tabla, que corresponde al signo "\$", tienes que contar posiciones de memoria en el archivo. Sabiendo que cada carácter son 8 bytes y que el dolar lo tienes en la posición 12 de la tabla, tenemos que pegar el contenido del portapapeles (que has debido conseguir haciendo previamente 'copy' de la selección de los bytes en el editor html. Ver la imagen inferior) posicionando el cursor de texto, con el ratón, en el primer carácter del byte número 88. El programa *wxHexEditor* te indicará, en la parte inferior, en todo momento, en qué lugar te encuentras posicionado, en el texto 'Desviación del cursor'.

Una vez pegado el contenido el aspecto es el que puedes ver en la página siguiente...



1818181818001800

charset_videopac.chr																																																				
Posici	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	
000000	7C	C6	C6	C6	C6	C6	7C	00	18	38	18	18	18	18	3C	00	3C	66	0C	18	30	60	7E	00	7C	C6	06	3C	06	C6	7C	00	CC	CC	CC	FE	0C	0C	00	FE	C0	C0	7C	06	C6	7C	00	7C	C6	C0		
000051	FC	C6	C6	7C	00	FE	06	0C	18	30	60	C0	C0	7C	C6	7C	C6	C6	7C	00	7C	C6	7E	06	C6	7C	00	00	18	18	00	18	00	00	18	18	18	18	00	18	00	00	00	00	00	00	00	00	00	00	00	00
000102	00	00	3C	66	0C	18	18	00	18	00	C0	C0	C0	C0	C0	FE	00	FC	C6	C6	FC	C0	C0	C0	00	00	18	18	7E	18	18	00	00	C6	C6	C6	D6	FE	EE	C6	00	FE	C0	C0	F8	C0	C0	FE	00	FC		
000153	C6	C6	FC	D8	CC	C6	00	7E	18	18	18	18	18	00	C6	C6	C6	C6	C6	C6	7C	00	3C	18	18	18	18	18	3C	00	7C	C6	C6	C6	C6	C6	7C	00	7C	C6	C6	DE	CC	76	00	7C	C6	C0	7C			
000204	06	C6	7C	00	FC	C6	C6	C6	C6	C6	FC	00	FE	C0	F8	C0	C0	C0	00	7C	C6	C0	00	CE	C6	7E	00	C6	C6	FE	C6	C6	C6	00	06	06	06	06	C6	7C	00	C6	CC	D8	F0	D8	CC	C6				
000255	00	38	6C	C6	C6	FE	C6	C6	00	7E	06	0C	18	30	60	7E	00	C6	C6	6C	38	6C	C6	C6	00	7C	C6	C0	C0	C6	7C	00	C6	C6	C6	C6	6C	38	00	FC	C6	C6	FC	C6	FC	00	C6	EE				
000306	FE	D6	C6	C6	C6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	66	3C	18	3C	66	00	00	18	00	7E	00	18	00	00	00	00	00	7C	00	7C	00	00	66	66	66	3C	18			
000357	18	18	00	C6	E6	F6	FE	DE	CE	C6	00	03	06	0C	18	30	60	C0	00	FF	FF	FF	FF	FF	FF	00	CE	DB	DB	DB	DB	CE	00	00	00	3C	7E	7E	3C	00	1C	18	1E	18	18	1C	00					
000408	1C	1C	18	1E	18	34	26	00	38	38	18	78	18	2C	64	00	38	38	18	78	18	18	38	00	00	18	0C	FE	0C	18	00	18	3C	7E	FF	FF	18	18	00	03	07	0F	1F	3F	7F	FF	00	C0	E0	F0		
000459	F8	FC	FE	FF	00	38	38	12	FE	B8	28	6C	00	C0	60	30	18	0C	06	03	00	00	00	0C	08	FF	7E	00	00	03	63	FF	FF	18	08	00	00	00	00	10	38	FF	7E	00	00	00	06	6E	FF			
000510	7E	00																																																		
000561																																																				

Debes repetir la misma operación con cada carácter que deseas cambiar, hasta que te quede un conjunto de datos como este...

charset_videopac.chr																																																				
Posici	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	
000000	7C	C6	C6	C6	C6	C6	7C	00	18	38	18	18	18	18	3C	00	3C	66	0C	18	30	60	7E	00	7C	C6	06	3C	06	C6	7C	00	CC	CC	CC	FE	0C	0C	00	FE	C0	C0	7C	06	C6	7C	00	7C	C6	C0		
000051	FC	C6	C6	7C	00	FE	06	0C	18	30	60	C0	C0	7C	C6	7C	C6	C6	7C	00	7C	C6	7E	06	C6	7C	00	00	18	18	00	18	00	00	18	18	18	18	00	18	00	00	00	00	00	00	00	00	00	00	00	00
000102	00	00	3C	66	0C	18	18	00	18	00	C0	C0	C0	C0	C0	FE	00	FC	C6	C6	FC	C0	C0	C0	00	00	18	18	7E	18	18	00	00	C6	C6	C6	D6	FE	EE	C6	00	FE	C0	C0	F8	C0	C0	FE	00	FC		
000153	C6	C6	FC	D8	CC	C6	00	7E	18	18	18	18	18	00	C6	C6	C6	C6	C6	C6	7C	00	3C	18	18	18	18	18	3C	00	7C	00	E6	F6	DE	CE	C6	00	7C	C6	C6	DE	CC	76	00	7C	C6	C0	7C			
000204	06	C6	7C	00	FC	C6	C6	C6	C6	C6	FC	00	FE	C0	F8	C0	C0	C0	00	7C	C6	C0	00	CE	C6	7E	00	C6	C6	FE	C6	C6	C6	00	06	06	06	06	C6	7C	00	C6	CC	D8	F0	D8	CC	C6				
000255	00	38	6C	C6	C6	FE	C6	C6	00	7E	06	0C	18	30	60	7E	00	C6	C6	6C	38	6C	C6	C6	00	7C	C6	C0	C0	C6	7C	00	C6	C6	C6	C6	C6	6C	38	00	FC	C6	C6	FC	C6	FC	00	C6	EE			
000306	FE	D6	C6	C6	C6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	66	3C	18	3C	66	00	00	00	06	0C	18	30	60	00	00	00	00	7C	00	7C	00	00	66	66	66	3C	18				
000357	18	18	00	C6	E6	F6	FE	DE	CE	C6	00	03	06	0C	18	30	66	C6	00	FF	FF	FF	FF	FF	FF	00	92	54	38	EE	38	54	92	00	00	00	3C	7E	7E	3C	00	1C	18	1E	18	18	1C	00				
000408	1C	1C	18	1E	18	34	26	00	38	38	18	78	18	2C	64	00	38	38	18	78	18	18	38	00	00	18	0C	FE	0C	18	00	18	3C	7E	FF	FF	18	18	00	03	07	0F	1F	3F	7F	FF	00	C0	E0	F0		
000459	F8	FC	FE	FF	00	38	38	12	FE	B8	28	6C	00	00	00	00	00	08	08	10	00	00	00	0C	08	FF	7E	00	00	03	63	FF	FF	18	08	00	00	00	00	10	38	FF	7E	00	00	00	06	6E	FF			
000510	7E	00																																																		
000561																																																				

Ahora solo tienes que guardarlo con el nombre "*mi_charset.chr*" en la tarjeta microSD de tu FPGA y ya estará listo para usar!

Veamos el resultado con este caso de estudio, el clásico juego Videopac N°22, -Monstruo Espacial- que representa un buen ejemplo.

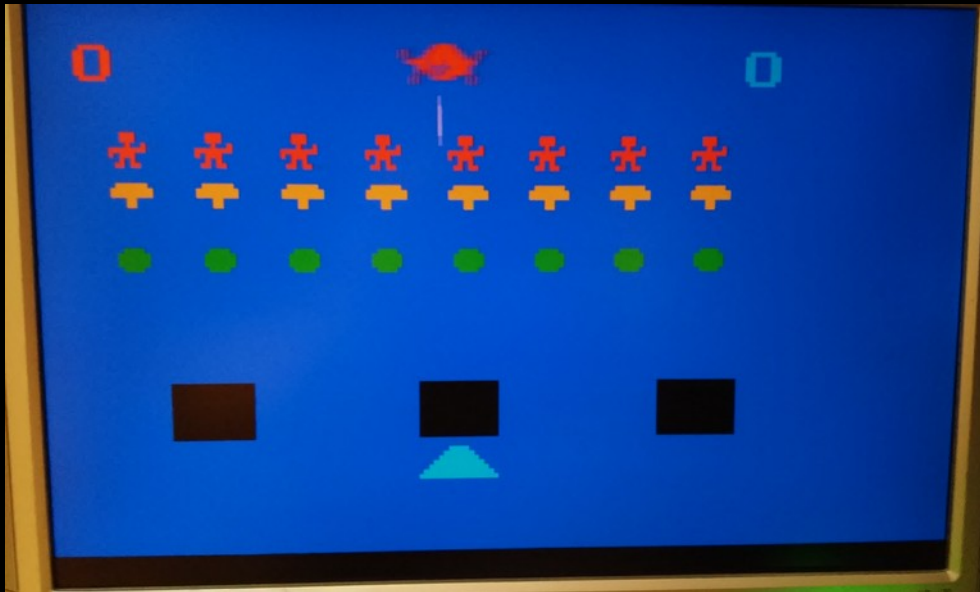


Necesitas 2 cosas:

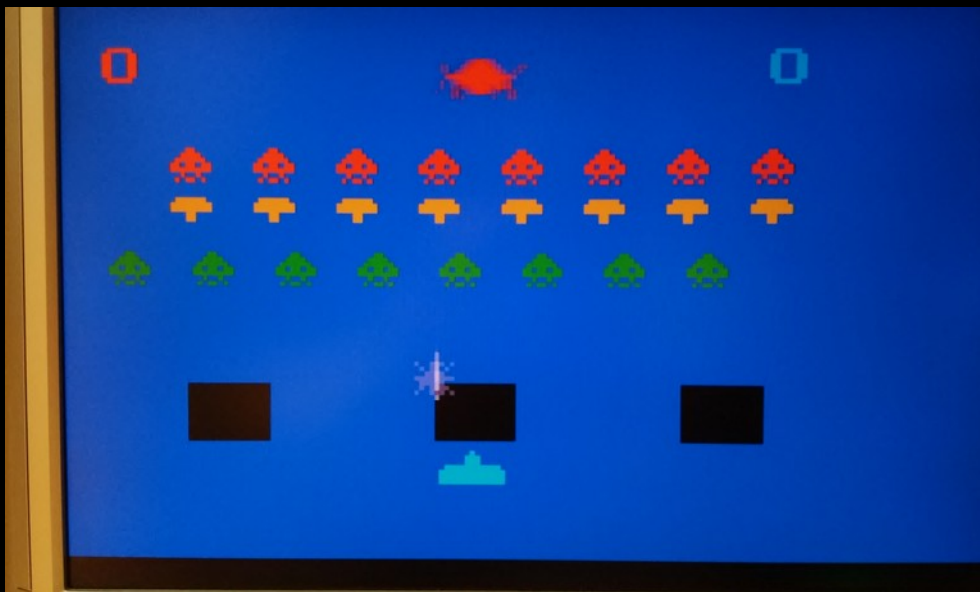
- El ROM del juego. Quizá puedas encontrarlo como "*vp_22.bin*" (2 KB)
- El archivo "*me_marcianitos.chr*" en (por ejemplo) la carpeta CHR, que tendrás creada dentro de la carpeta de juegos de Videopac.

Puedes cargar los 2 archivos en el orden que prefieras. Pero si cargas primero el juego para conocerlo y después cargas el archivo .chr podrás ver los cambios que se producen.

Primero lo verás así...



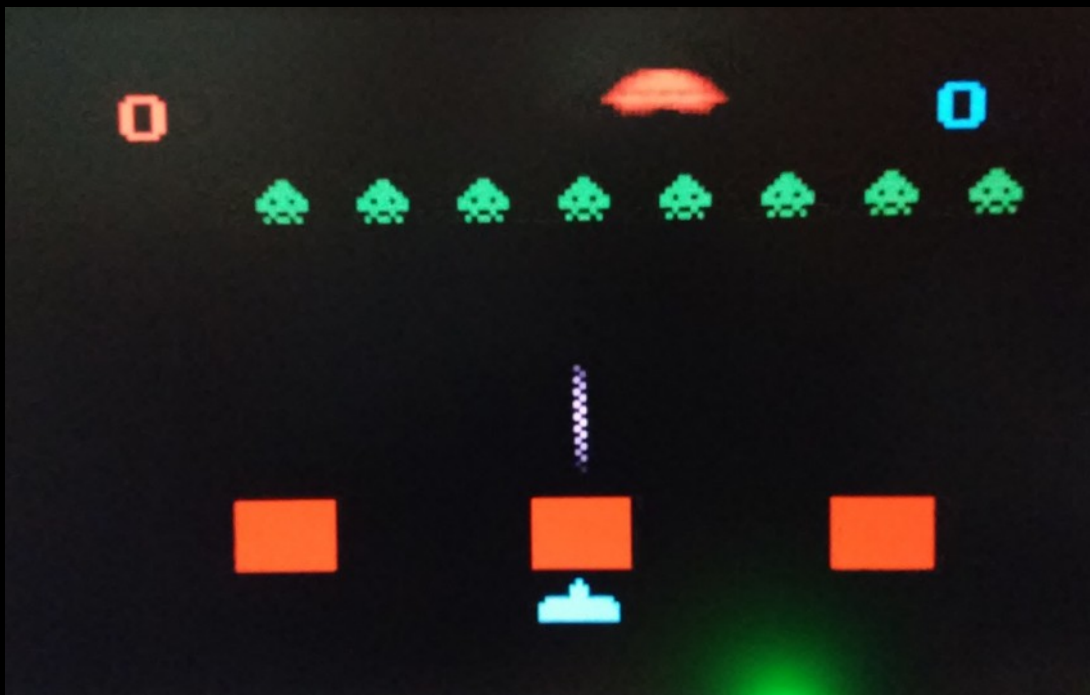
Y después así...



Puedes observar que los enemigos ha cambiado a terroríficos marcianos y que la pirámide que tenías como disparador de misiles ahora es un lanza misiles, como los de verdad ;)

Vamos un paso más allá:

Si ahora cargas el binario '*me_marcianitos.bin*' podrás ver que ha cambiado algo más.



Ahora han desaparecido los lanza misiles enemigos y sus bolas-barrera. También puedes ver que ya no tienes al monstruo espacial (en su lugar es una nave ovni) y que los misiles también tienen un aspecto diferente. Eso ocurre porque el charset no incluye (lógicamente) los sprites que cada juego pueda o no tener. Eso hay que cambiarlo directamente editando el binario del propio juego desde el *wxHexEditor*. No es una tarea sencilla dar con la posición en la memoria para cada uno de los gráficos.

La mecánica del juego es simple: Al primer disparo que alcanza al marciano le rompes su barrera de protección y al mismo tiempo ya no puede usar su arma láser. Al segundo disparo, el marciano, que estaba indefenso, queda eliminado.

Existe, en esta primera versión, un glitch conocido: La nave ovni se deforma al disparar contra nosotros.

Espero que este manual te resulte útil y que te animes a crear tus propios charsets para Videopac!!