Índice:

Capítulo	*Apartado*	*Página*
1. Una no tan pequeña introducción:		**2**
	1-1. *Abreviaciones de palabras.*	**2**
	1-2. *User Interface.*	**2**
	1-3. *Modo Visor.*	**3**
	1-4. *Modo Edición.*	**4**
	1-5. *Barra de Área.*	**4**
	1-6. *Info. sobre el Byte.*	**5**
	1-7. *Barra de UDG.*	**5**
2. Lista de teclas:		**6**
	2-1. *En el Modo Visor.*	**6**
	2-2. *En el Modo Edición.*	**7**
	2-3. *En el menú "Useful Locations".*	**8**
	2-4. *En la pantalla de MicroHobby Charsets.*	**9**
	2-5. *En los menús de Save y Load.*	**10**
3. Guardar y cargar datos, juegos de caracteres o de UDG manualmente.		**11**
4. Diferencias entre modelos, 48k y 128k.		**11**
5. Sobre los CharSets de MicroHobby.		**12**
6. Notas finales y agradecimientos.		**13**

1. Una no tan pequeña introducción:

1-1. Abreviaciones de palabros:

CShift -> Tecla '***Caps Shift***' del *Spectrum*.
SShift → Tecla '***Symbol Shift***' del *Spectrum*.
Char (o chars)-> Caracter/es (de los que se escriben, no que si es '*templao*' o no).
Charset → Set de caracteres.

1-2. User interface (View y Edit Mode):

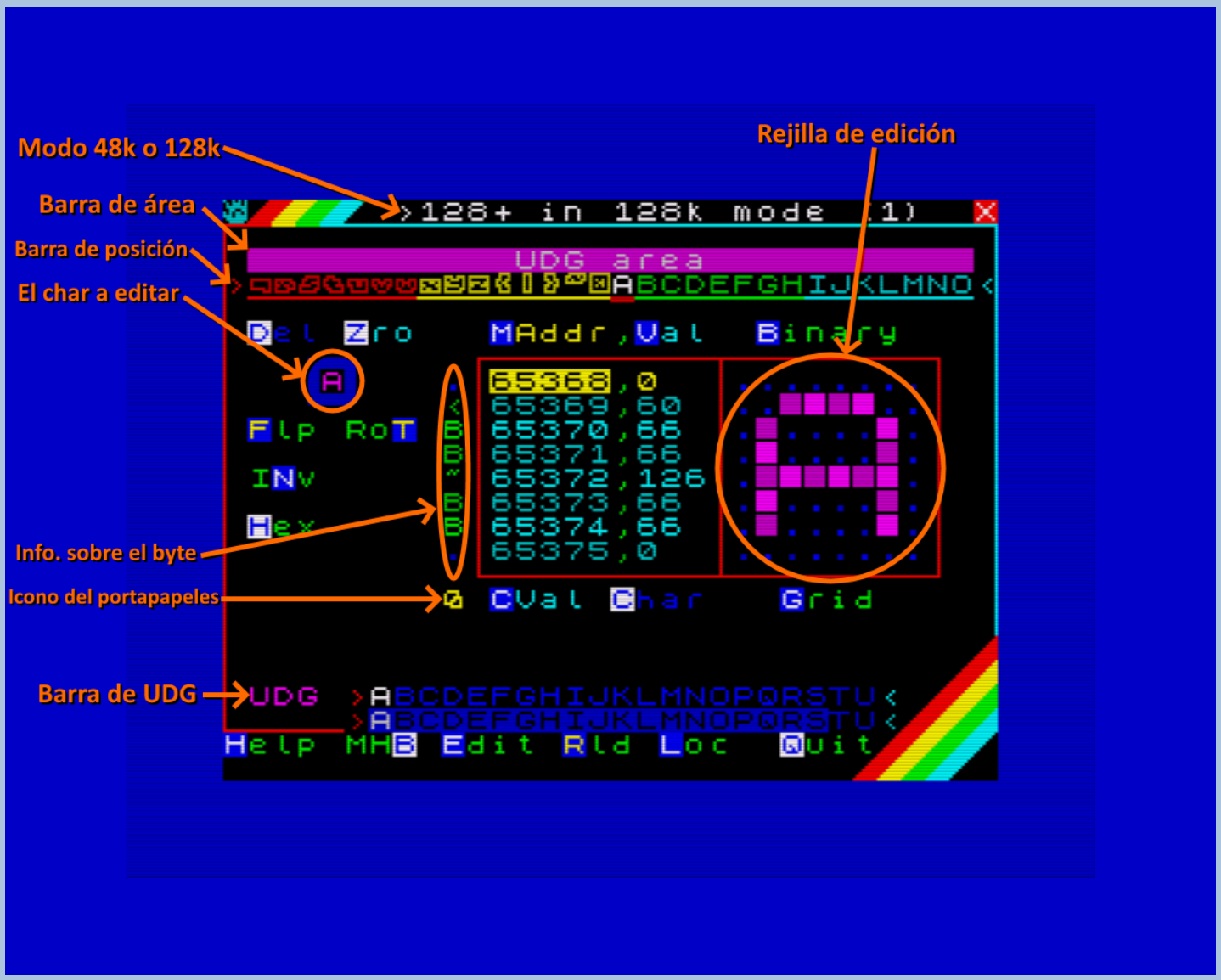


Imagen 1 – *Modo Visor (View Mode).*

Imagen 2 – *Modo Edición (Edit Mode).*

Usa estas imágenes para familiarizarte con la interfaz y sus diversos apartados y nombres.

1-3. Qué es el Modo Visor (View Mode):

Es el modo en el que empieza el programa, que sirve para -como su nombre bien indica- ver partes de la memoria, aunque se pueden hacer modificaciones básicas como *rotar, invertir, mirroring y más.*

En este modo te puedes desplazar por la totalidad de la memoria del *Spectrum*.

1-4. Qué es el Modo Edición (Edit Mode):

A este modo se entra pulsando la tecla 'E' o la tecla 'Enter' desde el ***Modo Visor***, y te permite modificar el char desde la rejilla directamente, como si fuese un editor de sprites. Sabrás que estás en ***Modo Edición*** porque el borde de la pantalla cambiará a color *Cyan*. Ni que decir que este modo es la salsa de este programa, si no se llamaría *Explorador de Caracteres*... un momento, ¡ups!

En este modo puedes hacer uso de la ***Barra de Sprites*** (CShift + S), que muestra agrupaciones de 2x2 chars para facilitar hacer sprites de 16x16 píxeles, y el modo ***Seamless Background*** (CShift + B), que te muestra una cuadrícula de 9x9 con el caracter que estés editando, para facilitarte hacer fondos que se repitan indefinidamente.

En este modo NO te puedes desplazar por toda la memoria, puesto que hay partes de ésta protegidas para que no se cause un crasheo del sistema por error. Nótese que si se está en el ***Modo Visor*** estando en una parte protegida de la memoria, no se podrá entrar al ***Modo Edición*** hasta que no se seleccione otra parte de la memoria que no esté protegida.

1-5. Barra de Área:

Las partes protegidas aparecen con un candado al final de la ***Barra de Área*** (la barra de color que aparece en la parte superior). Ver ***Imagen 3*** para mayor aclaración.

También el color de la ***Barra de Área*** indica si esa parte de la memoria está protegida o no:

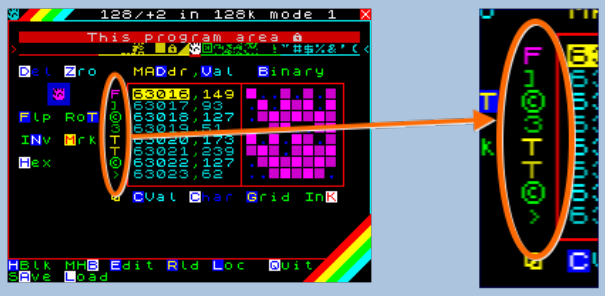
Imagen 3 – *Barra de Área.*

>**Azul oscuro**: Esta parte pertenece a la *ROM*, y **NO** se puede modificar.
>**Rojo**: Esta área pertenece al *BASIC*, este programa y otras cosas, y **NO** se puede modificar.
>**Magenta**: Esta área pertenece a los *UDG (Gráficos Definidos por el Usuario)* y **SÍ** se puede modificar.
>**Verde**: Esta área no está protegida y **SÍ** se puede modificar. Es la parte que deberías usar.
>**Cyan**: Esta área pertenece al *CharSet* de la *ROM* y evidentemente, **NO** se puede modificar.
>**Amarillo**: Pertenece a las *Variables del Sistema*, buffer de la impresora *(en 48k)*, etc. y **SÍ** se puede modificar, pero todavía estoy pensando que a lo mejor no es muy buena idea...
>**Blanco**: Pertenece a la memoria de video de la pantalla, y **NO** se puede modificar.

En resumen, sólo se permiten modificar las *áreas* que están en ***Verde***, ***Magenta*** *(¡mis ojos!)* y ***Amarillo***.

Estas restricciones también se aplican a las modificaciones que se pueden realizar en el ***Modo Visor*** (invertir, rotar, etc). O al menos deberían, cualquier bug pos ya sabéis, me haréis una persona muy feliz si me lo comunicáis. ;)

1-6. Leyenda de lo mostrado en la *Info. sobre el Byte:*



>Este apartado de la interfaz te proporciona información sobre el valor del byte de cada una de las ocho direcciones de memoria de la *rejilla.*

Te dice -si es un *código ASCII*- el caracter que representa, o si es un *token*, o si es un *código de control* especial usado por el *Spectrum* para controlar ciertas cosas.

Imagen 4 – *Info. sobre el Byte.*

El color del char que aparece en el apartado de ***Info. sobre el Byte*** (a la izquierda de las direcciones de memoria de la *rejilla*), te da información sobre el valor del byte de cada dirección de memoria. Te dice que *char* representa, si es un *token*, un *UDG*, etc.

>**Azul oscuro**: Aparece un **punto**. El valor del byte es menor de 32 y no representa ningún char del *ASCII*. Son *códigos de control* que usa el *Spectrum* (*cursores, teclas intro y borrar, colores*).

>**Verde**: El valor del byte está entre 32 y 143, y te dice que char del *código ASCII* representa.

>**Magenta**: El valor del byte está entre 144 hasta 162 -en *128k-*, y hasta 164 -en *48k-*, y representa los *UDG*. No aparece el gráfico que tengas definido, sólo la letra que lo representa.

>**Amarillo**: Aparece una '**T**'. El valor del byte está entre 162 -en *128k-*, o 164 -en *48k-* hasta 255, informando que representa un *token* del *Spectrum* (un comando del *BASIC, p. ej.*).

1-7. Barra de *UDG:*

>Esta barra de caracteres aparece cuando estás en la parte de la memoria donde residen los *UDG (a partir de 65 368).*

Imagen 5 - *Barra de UDG.*

Cuando entras a la parte de la memoria que contienen los *Gráficos Definidos por el Usuario (UDG)*, aparece esta barra para que tengas a golpe de vista todos los *UDG.*

2. La lista de teclas:

2-1. En el Modo Visor:

-*Desplazamiento DE la rejilla:*

>**O**:	–1 Char (–8 bytes).	>**P**:	+1 Char (+8 bytes).
>**CShift + O**:	–4 Chars (–32 bytes).	>**CShift + P**:	+4 Chars (+32 bytes).
>**Q**:	–4 Bytes.	>**A**:	+4 Bytes.
>**CShift + W**:	–1 Byte.	>**CShift + S**:	+1 Byte.
>**I**:	–8 Chars (–64 bytes).	>**K**:	+8 Chars (+64 bytes).
>**CShift + I**:	–1 Charset (–96 chars, –768 bytes.)	>**CShift + K**:	+1 Charset (+96 chars, +768 bytes).

>**W**: Mueve la ***barra de selección amarilla*** arriba.

>**S**: Mueve la ***barra de selección amarilla*** abajo.

-*General:*

>**B**: *Modo Binario.*
>**H**: Cambia de *decimal* a *hexadecimal* y viceversa.
>**E**, **Enter**: Entra al ***Modo Edición*** con el cursor en la *columna 4.*
>**R**: Refresca la *rejilla* y las *barras de posición* y de *UDG.*
>**CShift + R**: Recarga el programa.
>**L**: Entra en el menú '***Useful Locations***'.
>**CShift + Q**: Sale al BASIC (haz un **RUN** para volver al programa).
>**CShift + H**: Cambia el color del borde a negro (por si no te gusta el azul, ya ves).
>**CShift + B**: Pantalla de presentación y selección de *CharSets de* ***MicroHobby***.
>**D**: Cambiar a dirección de memoria (*Input*).

>**G**: Cambia el color de fondo de la *rejilla.*
>**CShift + G**: Cambia el color de los puntos de *fondo* (píxeles inactivos) de la *rejilla.*
>**SShift + K**: Cambia el color de la *tinta* (píxeles activos) de la *rejilla.*

>**CShift + A**: Entra al *menú de* ***Save****.*
>**CShift + L**: Entra al *menú de* ***Load****.*

>**M**: *Marca de selección* de dirección *inicial* de memoria para *copiar, mover, borrar* o *guardar (Save/Load).*

>**CShift + M**: *Marca de selección* de dirección *final* de memoria para *copiar, mover, borrar* o *guardar (Save).*

>**SShift + M** *(el punto)*, limpia las *marcas de selección/guardado* que hayas puesto.

-General (otras teclas):

>**SShift + Z**: Borra los datos entre las marcas de selección/guardado.
>**SShift + X**: Mueve los datos de las marcas de selección/guardado a la dirección actual.
>**SShift + V**: Copia los datos de las marcas de selección/guardado a la dirección actual.

>**Números** del **1** al **8**: Para entrar al ***Modo Edición*** con el cursor en el número de columna pulsado.

-Algo de edición:

>**V**: Cambiar el valor de la dirección de memoria seleccionada en amarillo (*Input*).
>**CShift + D**: Borra el char actual.
>**F**: Voltear el char verticalmente.
>**T**: Rotar en sentido horario.
>**N**: Invertir el char
>**CShift + Z**: Borra la fila (*byte*) seleccionado.
>**CShift + F**: Voltear el char horizontalmente.
>**CShift + T**: Rotar en sentido antihorario.

>**SShift + 6**, **7**, **8** y **9** *(Joystick Sinclair)*: 'Acomodar' el char (desplazarlo por la rejilla).

>**N**, **SShift + 0** *(Disparo en Joystick Sinclair)*: Invierte el char (un negativo, lo que es 0 ponerlo a 1 y viceversa).

-Portapapeles:

>**C**: Copia el valor (byte) de la selección.
>**CShift + C**: Copia el char (8 bytes) actual.
>**CShift + V**: Pega el valor (byte) o el char (8 bytes) en la posición actual.
>**CShift + X**: Borra el ***portapapeles***.

2-2. En el Modo Edición:

-Desplazamiento POR la rejilla:

>**W**, **A**, **S**, **D**: Mover cursor (*Arriba*, *Izquierda*, *Abajo* y *Derecha*).
>**O**, **P**: Mover cursor (*Izquierda*, *Derecha*).
>**Q**, **E**, **Z**, **C**: Mover cursor en diagonales (*Arriba izquierda*, *Arriba derecha*, *Abajo izquierda* y *Abajo derecha*).

>**CShift + O**, **P**: Guarda el char en memoria y se desplaza al char anterior/posterior.

-Escritura de píxeles:

>**X**: *Activa* la **escritura**. Se desactiva al volver a pulsar la tecla '**X**' o la **Barra Espaciadora**.
>**CShift + 0**, **Delete**: *Activa* el **borrador**. Se desactiva pulsando '**0**', '**CShift + 0**' o la **Barra Espaciadora**.
>**N**: *Activa el **píxel*** en la posición del cursor (incluso con la **escritura**/**borrador** activo).
>**M**: *Desactiva el **píxel*** en la posición del cursor (incluso con la **escritura**/**borrador** activo).
>**Barra Espaciadora**: *Activa/desactiva el **píxel*** en la posición del cursor.

-Modificaciones:

>**CShift + D**: Borra el char actual.
>**CShift + Z**: Borra la fila (byte) seleccionado en amarillo.
>**F**: Voltear el char verticalmente.
>**CShift + F**: Voltear el char horizontalmente.
>**T**: Rotar en sentido horario.
>**CShift + T**: Rotar en sentido antihorario.

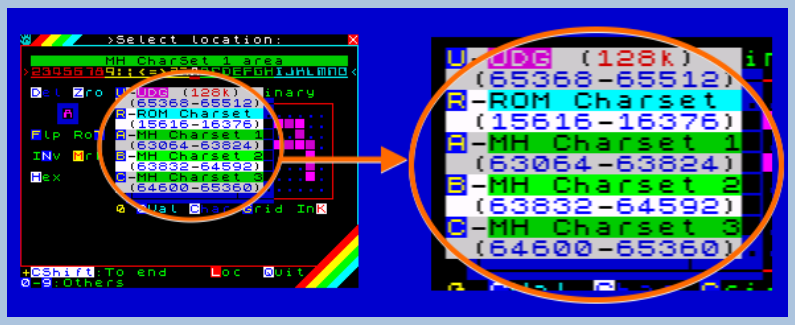
>**SShift + 6**, **7**, **8** y **9** *(Joystick Sinclair)*: 'Acomodar' el char (desplazarlo por la rejilla).

>**I**, **SShift + 0** *(Disparo en Joystick Sinclair)*: Invierte el char (lo mismo que la tecla '**N**' en el ***Modo Visor***).

-General:

>**B**: *Modo Binario.*
>**CShift + S**: Activa/desactiva ***modo Sprites***.
>**1**, **2** y **3**: *Offset* del ***modo Sprites*** (–1, +1 y reset a 0).
>**CShift + B**: Activa/desactiva el ***modo Seamless***, para facilitar dibujar *backgrounds* con repeticiones *sin costuras*, por ejemplo.
>**CShift + E**: ***NO*** *guarda* el char en memoria, restaura el original y vuelve al ***Modo Visor***.
>**Enter**: *Guarda* el char en memoria y vuelve al ***Modo Visor***.
>**CShift + Q**: Se *guarda* el char modificado en memoria y sale al *BASIC* (haz un *RUN* para volver al programa).
>**R**: Actualiza la *barra de posición*, la de *UDG* y la de *Sprites*.
>**CShift + R**: Recarga el char y todo lo demás (la rejilla y las *barras de posición*, *UDG* y *Sprites*).

2-3. En el menú de selección 'Useful Locations':



>En este menú puedes saltar rápido a direcciones de memoria predefinidas.

Imagen 6 *– Useful Locations menu.*
*(tecla '**L**' en el Modo Visor)*

>**U**: *UDG* (65 368 en 48k y 128k).
>**R**: *ROM* CharSet (15 616).
>**A**: *MicroHobby* CharSet 1 (63 064).
>**B**: *MicroHobby* CharSet 2 (63 832).
>**C**: *MicroHobby* CharSet 3 (64 600).
>**CShift + U**: Final de *UDG* (65 512 -en 128k-, 65 528 -en 48k-).
>**CShift + R**: Final de *ROM* CharSet (16 376).
>**CShift + A**: Final de *MicroHobby* CharSet 1 (63 824).
>**CShift + B**: Final de *MicroHobby* CharSet 2 (64 592).
>**CShift + C**: Final de *MicroHobby* CharSet 3 (65 360).
>**CShift + Q**: Sale al *BASIC* (haz un *RUN* para volver al programa).

– Lista de teclas. Modo Edición, Menú Useful Locations –

-Más teclas de 'Useful Locations':

>**1**: *Primer tercio* de la pantalla (16 384).
>**2**: *Segundo tercio* de la pantalla (18 432).
>**3**: *Tercer tercio* de la pantalla (20 480).
>**4**: *Atributos de color* de la pantalla (22 528).
>**5**: *Variables del Sistema* -en *128k*-, *búfer de la impresora* -en *48k*- (23 296). Las *variables del Sistema* en *48k* empiezan en 23 552.
>**6**: Comienzo de este programa (25 200).
>**7**: **Memoria de usuario libre (56 000).**
>**8**: Rutinas, datos y variables del programa (62 968).
>**9**: Rutina *MicroHobby CharSet* (63 040).
>**0**: Principio de la *ROM* (00 000).

2-4. En la pantalla de selección de Charsets de MicroHobby:

>En esta pantalla puedes cambiar la fuente de texto del programa por una de las tres que hay en el espacio de memoria dedicado a los Charsets de *MicroHobby*. Así puedes probarlos a ver que tal quedan.

Si tienes charsets en otra parte de la memoria y quieres probarlos, copia el charset a una de estas tres direcciones. Son las que corresponden a las teclas ***A***, ***B*** *y* ***C*** en el menú de *'Useful Locations'* (***CShift*** + ***L***).

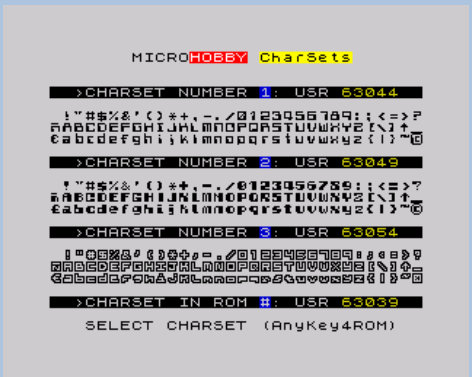


Imagen 7 – *MicroHobby CharSets.*
*(teclas '**CShift** + **B**' en el Modo Visor)*

*- Ve a la **página 12** para obtener más información sobre los Charsets de MicroHobby, y la parte de la memoria donde se alojan -*

-Ve a la sección 5 para obtener más información sobre los Charsets de MicroHobby, y la parte de la memoria donde se alojan-

>**1**: Selecciona el *Charset 1* y cambia la fuente del programa (para probar un nuevo *charset*, por ejemplo).
>**2**: Selecciona el *Charset 2* y lo mismo de arriba.
>**3**: Selecciona el *Charset 3* y...

>**Cualquier otra tecla** cambia a la fuente por defecto (*ROM*) y vuelve al ***Modo Visor***.

– Lista de teclas. Menú Useful Locations, MicroHobby Charsets –

2-5. En los menús de Save y Load:

Imagen 8 – *Menú de Save/Load.*

>Aquí puedes guardar/cargar tus *Charsets*, *UDG*, *Sprites*, etc.

Puedes guardar en *cinta* en todos los modelos de *Spectrum*, además de en *RAM* en los modelos de *128k* y también en *disco* en el ***+3***. También puedes guardar manualmente, saliendo al *BASIC* con **CShift + Q**.

Te recomiendo salir al *BASIC* antes y poner el *modo PANTALLA (SCREEN)* en un modelo de *128k*, para que no se borre la pantalla superior al guardar/cargar manualmente, y tener los datos a la vista.

>**1**: Guardar/Cargar *UDG's*.
>**2**: Guardar/Cargar *CharSets de **MicroHobby*** (los tres, 288 chars --2304 bytes sin rutina, --2329 bytes con rutina).
>**R**: Seleccionar guardar/cargar con o sin rutina.

>**3**: Guardar/Cargar *CharSet de **MicroHobby*** (sólo un charset, 96 chars --768 bytes).
>**M**: Seleccionar área para guardar/cargar *CharSet* (o con las teclas **A**, **B** ó **C**).

>**4**: Guardar/Cargar desde/hacia marca/s de selección/guardado.
>**5**: Guardar/Cargar desde/hacia dirección custom. Ha de ser *de 56000 ($DAC0) en adelante.*
>**6**: Guardar programa completo/Cargar desde file header.

>**T**: Guardar/Cargar en *Tape (cinta)*, *Disk* (disco) o *RAM*.

>**Con cualquier otra tecla** vuelves al ***Modo Visor***.

Ten cuidado al cargar, porque puedes cargar datos encima del programa y hacer que crashee. Si al guardar/cargar te da algún fallo o haces un **break**, vuelve a ejecutar el programa con un simple **RUN**.

Si necesitas grabar o cargar desde una unidad diferente (si tienes un ***divIDE***, o la *unidad **C*** del ***Spectrum Next***) debes hacerlo de forma manual saliendo al *BASIC* (**CShift + Q**).

Si piensas cargar charsets de la página de ***ZX Origins*** (*https://damieng.com/typography/zx-origins*) te recomiendo que los cargues en una de las tres áreas de los charsets de *MicroHobby* (la opción/tecla **3**).

– Lista de teclas. Menús de Save y Load –

3. Guardar y cargar datos, juegos de caracteres o de UDG manualmente:

Para grabar o cargar datos manualmente hay que salir al *BASIC* con las teclas **CShift + Q** (mejor te apuntas en un papelico la dirección de memoria y la longitud de los datos a grabar/cargar, o haces lo de salir al *BASIC* y seleccionar el *modo PANTALLA* en un modelo de *128k*).

Para cargar datos no hagas ningún *CLEAR xxxxx*. Si lo haces, posiblemente te cargues parte del código del programa y no puedas volver a él sin que casque. Ya hay un *CLEAR 25 199* en el cargador, que es donde empieza el programa. Desde donde acaba el programa (cerca de 56 000) hasta 65 535, es memoria libre que se puede usar (con excepción de una parte intermedia que usa el programa para unos pocos datos).

Mira muy bien dónde cargas las cosas para no pisar el código del programa. Recuerda que luego en tu programa/juego, puedes cargar estos datos en la posición de memoria que te resulte más cómoda sin problema alguno.

Puedes usar el mismo programa para ver que partes hay libres y así cargar datos sin miedo a machacar nada. En el ***Modo Visor***, usa **CShift + I** / **K** para moverte rápidamente con saltos de 768 bytes, y sin el **CShift** pulsado para saltos de 64 bytes.

4. Diferencias entre modelos, 48k y 128k:

El programa intenta detectar el modelo de *Spectrum*, y en qué modo está actualmente *(modo 128 ó 48)*. El programa también detecta si estás en un *+3* y usará la unidad de disco. En los modelos de *128k* es posible usar el disco *RAM*. En todos los modelos se puede guardar a cinta. Si no consigue detectar el modelo, lo tratará como un *Spectrum 128k* (los clones rusos, por ejemplo).

Según lo que detecte, se pone en un modo de 4 diferentes (en todos los modos se puede usar la cinta):

-**Modo 0**: Estás en un modelo de ***48k***, o en un modelo de ***128k*** en *modo 48* (*USR 0)*, y sólo se puede usar el *Tape (la cinta*).

-**Modo 1**: Estás en un ***128/+2*** en *modo 128*, o en un clon no detectado (se puede usar el disco *RAM*).

-**Modo 2**: Estás en un ***+2A/B*** en *modo 128* (se puede usar el disco *RAM*).

-**Modo 3**: Estás en ***+3*** o en un ***+2A/B*** con *unidad de disco* conectada (se puede usar el disco *RAM* o la *unidad de disco*).

¿En qué se diferencian estos modos? pues prácticamente sólo en la disposición del dispositivo de almacenamiento, pero también cambian ciertas cosicas puntuales que expongo a continuación:

En el menú de '***Useful Locations***', al pulsar la tecla **CShift + U** para ir al final de los *UDG*, en *128k* llega hasta el *UDG* **S**, y en el modo de *48k* llega hasta el *UDG* **U** (aunque se pueden editar la **T** y la **U** tanto en *48k* como en *128k*). En la ***Barra de Área***, en *48k* aparece el buffer de la impresora, en *128k* no.

5. Sobre los CharSets de MicroHobby:

El programa viene con tres sets de caracteres. Son los que venían en la revista *MicroHobby* número 30, páginas 14, 15 y 16, en un apartado llamado '*Personaliza tu Spectrum*', pero el tercero de ellos está cambiado por un set hecho por mi, y los otros dos con muy ligeros cambios en algún char. Los puedes usar, modificar a tu gusto o crear unos nuevos.

Aparte de los tres sets, en la revista venía una pequeña rutina en ensamblador para cambiar a cualquier set de los tres con un *RANDOMIZE USR*, que cambia la Variable del sistema *CHARS*. Esta rutina la he incluido.

En el programa, en el ***Modo Visor***, puedes pulsar **CShift + B** para ver los tres sets al completo, incluso seleccionar alguno que cambiará la fuente de texto del programa para ver como quedaría. En este menú de selección de charset puedes ver al lado de cada set un texto que dice '*USR xxxxx*', donde '*xxxxx*' es la dirección de memoria donde has de hacer el *RANDOMIZE USR* para cambiar a ese charset. Te lo pongo aquí también:

>**RANDOMIZE USR 63 044** para usar el primer set.
>**RANDOMIZE USR 63 049** para usar el segundo set.
>**RANDOMIZE USR 63 054** para usar el tercer set.
>**RANDOMIZE USR 63 039** para volver al set de caracteres de la *ROM*.

La rutina de selección de charset empieza en la dirección de memoria 63 039, y acaba en la 63 063 (inclusive), y ocupa tan sólo 25 bytes. Hay un programa que adjunto que crea la rutina para hasta 32 charsets.

Si usas estos sets de caracteres, te recomiendo que llames a la rutina del set de la *ROM* al salir del programa donde las uses, sobretodo en el *modo 128k* del *BASIC*, o veras basura (o directamente espacios en blanco) en vez del listado del programa. En tiempo de ejecución y en *modo 48k* se verán bien los caracteres. No pasa nada, ni el listado de tu programa se ha corrompido ni nada, es sólo la manera que tienen de funcionar los *Spectrum* de *128k*.

El primer charset va desde la dirección 63 064 (un charset completo son 768 bytes) hasta 63 831, el segundo desde 63 832 hasta 64 599, y el tercero desde 64 600 hasta 65 367 (justo hasta donde empiezan los *UDG*, que es 65 368). Ten en cuenta esto si sólo quieres guardar los charsets (o sólo alguno) sin la rutina de ensamblador antes dicha.

Si quieres más información sobre los charsets de *MicroHobby*, visita el siguiente enlace:

https://microhobby.speccy.cz/mhforever/numero030.htm

Busca el apartado '*Personaliza tu Spectrum*' (páginas 14, 15 y 16). Tienes incluso el código fuente de la rutina de ensamblador, que es muy sencilla, y la puedes modificar para añadir mas charsets, o para poder ubicar la rutina en otra dirección de memoria si te es necesario. Si es el caso, acuérdate de que la *variable del sistema CHARS* apunta a la dirección de memoria del juego de caracteres *MENOS 256*. O sea, que le has de decir a la variable *CHARS* que tienes el charset **256 bytes menos** que la posición donde realmente está el charset en memoria.

6. Notas finales y agradecimientos:

En la página de ***ZX Origins*** *(https://damieng.com/typography/zx-origins)* tienes a tu disposición cienes y cienes de charsets completos para descargar. Es oro la web esta. Muchas gracias a ***DamienG*** por la creación de la misma.

Si alguien sabe algún POKE o RANDOMIZE USR o algo para cambiar al *modo Screen* del *BASIC* de *128k* (*la pantalla inferior*), que por favor me diga cómo. Estaría de lujo que al salir del programa aparezca en este modo, para que así no se borre la pantalla superior y puedas tener, por ejemplo, la dirección a grabar a la vista.

Muchísimas gracias a ***vicentecno*** por su rutina de asm para averiguar *múltiplos de 32*, me ha sido muy útil para la *barra de Sprites*. Muchas gracias a ***Sergio thEpOpE*** por la sugerencia del *seamless background* y -sobre todo- por sus videos (me han ayudado mucho). Muchas gracias también a ***Rodolfo Guerra*** por descubrirme el compilador de ***HiSoft***.

Y muchas gracias sobretodo a ***vicentecno***, ***Paco Vespa***, ***Azimov***, y en general a toda la gente del discord de ***Arnau Jess*** por aguantarme (***Isaías***, ***Jimmy***), y en especial a la gente del canal ***#desarrollo*** del mismo servidor. Muchísimas gracias a todos...

- Hecho en ***Sinclair BASIC*** y compilado con ***HiSoft BASIC Compiler*** *(versión de 128k)* -

Fin

*·**PD:** Vale, puede que se me haya ido un poco de las manos tanta tecla. Lo siento mucho, no volverá a ocurrir (sonando con voz monárquica)...*