

# EL ALMA DEL SPECTRUM

## CAPITULO I

=====  
En la serie que comenzamos con este capítulo, vamos a adentrarnos en la forma de utilizar las rutinas en código máquina que se encuentran ocultas en el interior de la ROM. Lo que podríamos llamar: "EL ALMA DEL SPECTRUM".  
=====

Jesús Alonso Rodríguez

### INTRODUCCION:

Cuando conectamos el ordenador, y antes de pulsar ninguna tecla, podría parecer que está "parado". En realidad, aunque no estemos ejecutando ningún programa, el microprocesador está trabajando en todo momento, corriendo a una velocidad vertiginosa y realizando más de medio millón de operaciones por segundo.

Pero, si no le hemos cargado ningún programa en Basic, ¿que programa está ejecutando el microprocesador?. Para contestar esta pregunta es necesario comprender qué es un ordenador.

### HARDWARE Y SOFTWARE:

Si nos contemplamos a nosotros mismos como seres humanos, podemos percibir que además de nuestro cuerpo, la parte física y tangible, existe algo más, una parte inmaterial que denominamos "alma" o "mente".

A su nivel, en un ordenador podemos distinguir también dos partes fundamentales, una de ellas está compuesta por los circuitos electrónicos que realizan las diferentes funciones, a esta parte la denominamos "Hardware" o "Soporte Físico". La parte inmaterial es el conjunto de información almacenada en la memoria, que controla las actuaciones del hardware y a la que denominamos "Software" o "Soporte Lógico".

Los programas en Basic que introducimos en nuestro ordenador forman parte del software, pero el microprocesador es incapaz de entenderlos. El eslabón intermedio entre nosotros y el microprocesador es el Intérprete de Basic.

### EL SISTEMA OPERATIVO:

Cuando el ordenador sale de fábrica, trae incorporado parte del software que utilizará para trabajar. Este software está compuesto por una serie de programas escritos en un lenguaje que sí entiende el microprocesador -Código Máquina- y que se encuentran en una memoria que permanece inalterada aunque se apague el ordenador, la memoria ROM.

A este conjunto de programas le denominamos "Sistema Operativo" y contiene el Editor (que nos sirve para escribir nuestros programas), el Intérprete de Basic (que los ejecuta) y una serie de rutinas auxiliares que permiten a todo esto funcionar.

A lo largo de esta serie, vamos a aprender la forma de utilizar estas rutinas desde nuestros programas, bien para hacer cosas que no se pueden hacer en Basic, bien para ayudarnos en la confección de programas en código máquina o simplemente, para que nuestros programas corran más deprisa.

### ESTRUCTURA DE LA ROM:

Dado que el microprocesador empieza a funcionar desde el mismo momento que se conecta, lo primero que encontramos en la ROM son una serie de rutinas que "inicializan" el ordenador, es decir, que fijan una serie de parámetros para que las sucesivas rutinas puedan funcionar.

Hay otra rutina muy importante, que se encarga de leer los datos que introducimos por el teclado y nos permite comunicarnos con el ordenador, el microprocesador interrumpe lo que está haciendo cada 20 milisegundos, y ejecuta esta rutina para saber si le hemos introducido algún dato nuevo a través del teclado.

Finalmente, está la "Rutina Principal de Ejecución" que se encarga de comprobar si las líneas en Basic que introducimos son correctas, de almacenarlas en memoria y de ejecutarlas, llamando para ello a una serie de subrutinas auxiliares que van ejecutando los distintos comandos del Basic.

Hay rutinas que transfieren bloques, rutinas que mandan datos por los distintos canales de comunicación, rutinas que controlan la pantalla, algunas realizan cálculos, otras controlan periféricos, y a todas podemos acceder y usarlas para nuestros fines, si sabemos donde están y como entrar en ellas.

### ENTRAR EN LA ROM:

El intérprete de Basic está constituido por un gran número de pequeñas rutinas en Código Máquina que la rutina principal va llamando sucesivamente para ejecutar los distintos comandos.

En algunas de estas rutinas, se puede entrar directamente desde la dirección de comienzo, otras admiten entradas alternativas, y finalmente, algunas requieren determinadas condiciones de entrada, tales como un valor dado en un registro, una posición de memoria, etc.

De todos es sabido que para ejecutar una rutina en Código Máquina desde el Basic se utiliza la función USR seguida de un argumento numérico que es, precisamente, la dirección de

comienzo de la rutina y que dá como resultado el valor que contenía el registro doble BC cuando el microprocesador retornó desde el Código Máquina.

En las rutinas que no requieran condiciones previas de entrada, se podrá hacer RANDOMIZE USR a la dirección de comienzo. Cuando haya que entrar en la rutina con un determinado valor en algunos registros, se diseñará un pequeño programa en Código Máquina que se cargará en el Buffer de impresora mediante un cargador en Basic, con lo cual, estas rutinas podrán ser utilizadas incluso por aquellos que no sepan absolutamente nada de Código Máquina.

Para permitir al lector utilizar en sus propios programas más de una de estas rutinas, se escribirán de forma reubicable para que se puedan colocar una a continuación de otra. La forma de ejecutarlas desde el Basic será hacer RANDOMIZE USR seguido de la dirección de comienzo que, en este caso, corresponderá a una dirección del Buffer de impresora.

Cuando estas rutinas o las de la ROM se ejecuten en un programa que utilice la función RND, el uso de RANDOMIZE puede alterar la secuencia aleatoria, en ese caso será mejor utilizar otro comando antes de USR, por ejemplo: LET z=USR (dirección).

#### ESTRUCTURA DE LA SERIE:

En cada capítulo trataremos una o varias subrutinas de la ROM, llevando un orden lógico que nos permita aproximarnos con facilidad al conocimiento del sistema operativo.

Para cada rutina habrá una serie de apartados fijos, que serán: Función, Descripción, Entradas alternativas, Condiciones de entrada, Registros empleados, Condiciones de salida y Posibles utilidades; complementados con los que fueran necesarios en cada caso, así como figuras, tablas o listados, cuando se requieran para una mejor comprensión.

En cuanto a la nomenclatura, utilizaremos la propia del Assembler, los nombres de las variables serán los utilizados en el manual, y las etiquetas las correspondientes al código fuente del Sistema.

Los números se representarán en hexadecimal, indicándose con una "h." al final del número; a continuación irá ese mismo número en decimal, con una "d." para indicarlo; en este segundo caso, se utilizará la notación americana de "coma" para separar los "millares" y "punto" para los "decimales".

Por ejemplo: La dirección de memoria "cuatro mil quinientos treinta y cinco" (correspondiente a la rutina del comando NEW), se representará: 11B7h. (4,535d.)

#### LOS REINICIOS (RST) DE PAGINA CERO:

Existen una serie de instrucciones que fuerzan al microprocesador a saltar a una subrutina que empieza en una de las primeras direcciones de memoria, cada una de estas instrucciones solo ocupan un byte (frente a los tres de una

llamada a subrutina) por lo que se utilizan para las subrutinas de uso más frecuente. Se llaman: RST 0, RST 8, RST 10, RST 18, RST 20, RST 28, RST 30 y RST 38. En ellas los programadores de Sinclair han colocado las siguientes rutinas:

RST 0 : Inicialización (START)  
RST 8 : Reinicio de error (ERROR-1)  
RST 10: Rutina general de salida (PRINT-OUT)  
RST 18: Recoge un caracter (GET-CHAR)  
RST 20: Recoge siguiente caracter (NEXT-CHAR)  
RST 28: Entrada al calculador (FP-CALC)  
RST 30: Hacer BC espacios (BC-SPACES)  
RST 38: Rutina de la interrupción enmascarable (MASK-INT)

Empezaremos nuestro estudio de la ROM por estas rutinas, y concretamente, en el próximo capítulo veremos las rutinas de inicialización (START).

----- 0 -----

# EL ALMA DEL SPECTRUM

## CAPITULO II

### LA INICIALIZACION DEL SISTEMA

En este segundo capitulo de nuestra serie, vamos a estudiar las rutinas que fijan los parámetros iniciales para que el Sistema empiece a trabajar.

Jesús Alonso Rodríguez

#### FUNCION:

En el momento de conectar el ordenador, el contenido de las posiciones de memoria y de los registros del microprocesador es aleatorio. En ese instante, el microprocesador recibe un impulso eléctrico de RESET que pone a cero todos sus registros, incluido el "PC" (Contador de Programa), por lo que empieza a correr desde la posición de memoria "cero".

Antes de saltar a la rutina principal de ejecución, el microprocesador debe borrar toda la memoria, y fijar el valor inicial de las variables del sistema y otros parámetros que utilizará posteriormente. El conjunto de rutinas que realizan estas funciones, se denomina "Inicialización del Sistema".

Todos los sistemas informáticos requieren una inicialización, en el caso del Spectrum, estas rutinas tienen, además, la finalidad de comprobar la memoria disponible, que será distinta según se trate de una versión de 16 o 48K, o tenga conectado un interface cuyo Sistema Operativo ocupe parte de la memoria.

#### DESCRIPCION:

La mayor parte de la rutina de inicialización se utiliza también cuando se ejecuta un comando NEW, esta parte se denomina "START/NEW" y comienza en la dirección 11C8h. (4,555d.). Esta rutina se comporta de forma distinta en función del contenido del registro "A" ("00" para START, "FF" para NEW).

Veremos primero la rutina START, luego la NEW, y finalmente, la START/NEW.

#### RUTINA "START":

La misión de esta rutina, que empieza en la dirección "cero", es fijar las condiciones de entrada en START/NEW para que se produzca una inicialización total del sistema.

Desactiva la interrupción enmascarable, pone a "cero" el registro "A", pone a FFFFh. (65,535d.) el registro "DE" y finalmente, salta a la rutina "START/NEW".

#### RUTINA "NEW":

Se trata de la rutina correspondiente a la ejecución del comando NEW, no se ejecuta al conectar el ordenador, pero se la puede considerar parte de la inicialización del sistema, de hecho hace el papel de un "WARM RESET" (Arranque en "caliente").

Comienza en la dirección 11B7h. (4,535d.) y su misión es fijar las condiciones de entrada en START/NEW para que se produzca la inicialización parcial correspondiente a la ejecución del comando NEW.

Desactiva la interrupción enmascarable, pone a FFh. (255d.) el registro "A", copia en el registro "DE" el valor de la variable RANTOP, Salva en los registros alternativos las variables P-RANT, RASP, PIP y UDG y salta a la rutina "START/NEW".

#### RUTINA "START/NEW":

Esta es la rutina principal de inicialización, común para START y NEW. Su función es producir la inicialización total o parcial del sistema, en función del contenido del registro "A".

Pone en blanco el color del Borde, chequea la memoria disponible y la pone a "cero", fija los valores iniciales de las variables del sistema, fija la posición de las distintas "pilas" (máquina, GO SUB y calculador) y el valor del registro índice "IY" que se utilizará para acceder a las variables del sistema, copia las 21 primeras letras mayúsculas en la zona de los UDG (esto último, solo si se entra desde START), fija los valores iniciales de los canales de comunicación, limpia el buffer de impresora, borra la pantalla y finalmente, imprime el conocido mensaje: "(c) 1982 Sinclair Research Ltd.", saltando a la rutina principal de ejecución después de volver a activar la interrupción enmascarable.

#### ENTRADAS UTILES:

Existen dos entradas principales en estas rutinas, una es la dirección "0" (RANDOMIZE USR 0) que tiene el mismo efecto que desconectar y volver a conectar el aparato o pulsar el RESET (si se dispone de él), otra es la dirección 11B7h. (4,535d.), RANDOMIZE USR 4535 tendrá el mismo efecto que NEW.

Se pueden conseguir efectos curiosos entrando por determinados puntos en la rutina START/NEW, RANDOMIZE USR 4709 puede asustarnos pensando que se ha producido un borrado total, en realidad, simplemente se fijan los colores, las variables REPPER, REPDEL, KSTATE-0, KSTATE-4, STRMS y DF-SZ, se limpia el buffer de impresora y se borra la pantalla, imprimiéndose el mensaje de Sinclair, no obstante, el programa que tuviéramos en memoria no se borra. Puede constatarlo haciendo:

```
10 REM Comprobacion
RANDOMIZE USR 4709
LIST
```

Verá que la línea 10 sigue existiendo. Este efecto permite algunas entradas alternativas, RANDOMIZE USR 4746 hace lo mismo, pero no cambia los colores que tuviéramos definidos en INK, PAPER y BORDER. RANDOMIZE USR 4754, hace lo mismo, pero no borra

el buffer de impresora, y RANDOMIZE USR 4757 se limita a imprimir el mensaje de Sinclair sin ni siquiera borrar la pantalla.

#### LOS CANALES DE COMUNICACION:

Uno de los datos mas interesantes que puede extraerse del estudio de estas rutinas, es la disposicion de los canales de comunicacion.

En la version básica del Spectrum existen cuatro canales, se denominan "K", "R", "S" y "P"; cada uno de ellos puede trabajar como entrada o como salida y tres de ellos estan asociados a los "streams" (corrientes) números #1, #2 y #3. De momento ignoraremos el canal "R" ya que se refiere al area de trabajo, y no se maneja con los "streams".

El canal "K" utilizado como salida, imprime en la parte inferior de la pantalla y utilizado como entrada, lee datos del teclado. El canal "S" utilizado como salida, imprime en la parte superior de la pantalla y utilizado como entrada, dá el informe: "J Invalid I/O device" (Dispositivo de Entrada/Salida no válido). Finalmente, el canal "P" maneja la impresora como salida, y dá el mismo informe de error como entrada.

El canal "K" está asociado con el "stream" #1, el canal "S" con el "stream" #2 y el canal "P" con el "stream" #3. De forma que PRINT #1 imprime en la parte inferior de la pantalla, PRINT #2 es equivalente a PRINT (escribe en la parte superior de la pantalla) y PRINT #3 es equivalente a LPRINT, es decir, escribe en la impresora. De la misma forma, INPUT #1, es equivalente a INPUT, es decir, escribe en la parte inferior de la pantalla y lee datos del teclado, pero ¿que pasa con INPUT #2 e INPUT #3 ?

Hemos dicho que los canales "S" y "P" no valen como entradas, de forma que nos saldrá el informe "J Invalid I/O device" pero la estructura de los canales depende de unos datos que son almacenados en el "area de informacion para canales" durante la rutina de inicializacion, si cambiamos estos datos, podremos hacer un INPUT en la parte superior de la pantalla, e incluso en la impresora.

En la TABLA 1 se puede ver como queda el area de informacion para canales, despues de la inicializacion. Si cambiamos algunos datos, podremos experimentar algunos efectos curiosos; teclee:

```
POKE 23741,168: POKE 23742,16
INPUT #2;"Dime tu nombre":a$
```

Verá que ahora puede hacer un INPUT en la parte superior de la pantalla. La parte de la sentencia que se imprime (en este caso el literal "Dime tu nombre") puede dirigirla a cualquier lugar de la pantalla, pero la entrada de datos será siempre por la segunda linea. Puede experimentar otras posibilidades, si dispone de impresora, ejecute el siguiente ejemplo que resulta muy ilustrativo sobre la forma de trabajar de INPUT:

```
POKE 23751,168: POKE 23752,16
INPUT #3;"Dime tu nombre":a$
```

Parece que no ha pasado nada, pero ahora, teclee su nombre, pulse ENTER con lo que recobrará el control, y finalmente, pulse LPRINT (y ENTER) para vaciar el buffer. Si el nombre tecleado fuera "Pepe" deberá ver en la impresora lo siguiente:

```
Dime tu nombre""P""Pe""Pep""Pepe"
```

Lo que demuestra que los datos han sido enviados a la impresora de la misma forma que lo serian a la parte inferior de la pantalla en un INPUT normal.

Dejamos a la imaginacion de los lectores las aplicaciones concretas, y esperamos que nos remitan un gran número de trucos basados en cambiar las direcciones del area de informacion para canales.

La semana próxima veremos los puntos de entrada a las rutinas de uso mas frecuente, y aprenderemos, entre otras cosas, a imprimir por nuestra cuenta todos los mensajes que normalmente nos dá el ordenador.

----- o -----

TABLA I

<u>DIRECCIONES</u>	<u>DATOS</u>	<u>SIGNIFICADO</u>
23734 y 35	244,9	Direccion 2548d. (Rutina PRINT-OUT)
23736 y 37	168,16	Direccion 4264d. (Rutina KEY-INPUT)
23738	75	Código "K" (Canal "K")
23739 y 40	244,9	Direccion 2548d. (Rutina PRINT-OUT)
23741 y 42	196,21	Direccion 5572d. (Rutina REPORT-J)
23743	83	Código "S" (Canal "S")
23744 y 45	129,15	Direccion 3969d. (Rutina ADD-CHAR)
23746 y 47	196,21	Direccion 5572d. (Rutina REPORT-J)
23748	82	Código "R" (Canal "R")
23749 y 50	244,9	Direccion 2548d. (Rutina PRINT-OUT)
23751 y 52	196,21	Direccion 5572d. (Rutina REPORT-J)
23753	80	Código "P" (Canal "P")
23754	128	Marca de "final de tabla"