Swift Disk

El Swift Disk fue creado por la compañía Petron Electronics. Ofrece una combinación facil de usar entre la interface de discos y el botón de copiado como el Multiface.
La interface totalmente compatible con el 48k y el 128k, tiene conectores para un joystick Kempston, puerto serial para impresora y la posibilidad de conectar hasta cuatro disqueteras, incluido un botón para acceder al modo DOS.
La versión 1 era óptima para los amantes de los juegos, pero limitado para usuarios más serios y programadores. La versión 2 en cambio, añade comandos basic y DOS, siendo la sintaxis un poco más fácil que la del Microdrive. La versión 3 añade a todo esto la posibilidad de emular al IF1 con Microdrives.

SwiftDisk II
El Swiftdisk II fue creado por la compañía Sixwords Ltd, el cual permite usar gran variedad de disqueteras de 3.5" y es una versión mejorada del anterior Swiftdisk. A diferencia de su predecesor que se conectaba en la parte baja del Spectrum, éste encaja más perfectamente y permite a su vez el uso de otros periféricos.
Se hicieron varias configuraciones, todas opcionales, del Swiftdisk II, pudiendo elegirse entre una con toma serial o con toma paralelo para conectar impresoras, además del "Microdrive Hardware Mapping", para trabajar en modo Microdrive emulado.

Hay tres modos de trabajar con el Swiftdisk:
 a) modo SwiftDos - basta pulsar el botón de la interface para entrar en esta modalidad.
 b) desde el ZX-basic - basta con añadir un "%" al comando Basic de disco, para diferenciarlo del casete o microdrive.
 c) modo Microdrive - no necesita Interface 1, basta cargar el emulador para poder usar el disco como un Microdrive.

sintaxis: comando [%][#flujo,];"nombre"[,parámetros]

Comandos del Swift DOS:
Format, Load, Save, Move, Erase, Catalogue, Date, Alter, Backup, Keep, Unkeep, Quit, *.

Comandos del Swift Basic:
Format%, Cat%, Load%, Save%, Merge%, Erase%, Open#%, Close#%, Print#, Input#, Inkey$#, In%, Out%, %eof, Lprint, Llist.

Los drives admitidos son numerados del 0 al 3.
El número de drive puede ser un número, una variable numérica o alfanumérica con Val.
El % indica que se usará el disco en vez del casete.
El nombre es de hasta 10 caracteres.

Swift DOS:
Bajo esta modalidad los comandos son tipeados letra a letra. El DOS admite incluso la primera letra de un comando, así basta tipear L por LOAD. Si uno olvida añadir los parámetros requeridos, el sistema consultará al usuario para obtener toda la información necesaria.

Format

sintaxis: F / FORMAT
donde:
nombre = un texto de hasta 10 caracteres
fichas = número máximo de archivos a contener (32 por defecto)
Este comando formatea y da nombre al volumen de los discos. Toma dos minutos formatear un disco a 640k. A mayor número de fichas, mayor lentitud y devoramiento de espacio en el disco.

Catalogue
sintaxis: C / CATALOGUE
donde:
B = Brief (breve)
F = Full (completo)
CATALOGUE muestra los datos del directorio en dos formas a elegir: uno breve y otro completo. El primero solo lista los nombres de cada archivo, la fecha de creación, el tamaño en sectores de 256 caracteres y el tipo de archivo. El completo, añade a eso las direcciones del código, su posición de comienzo o el tamaño de los registros por fichero.

Save
Sintaxis: S / SAVE [] [ ]
donde:
d = un valor del 0 al 3
b = el color del borde seleccionado para verlo al cargar lo salvado
S = SCREEN$ - salva solo la pantalla
SAVE permite salvar imágenes completas de programas en memoria, pantallas, códigos o datos.
Ejemplo:
SAVE 0;FRED 7 - salvará el contenido entero de la memoria al disco cero bajo el nombre FRED.
Los datos no se guardan comprimidos como con el Multiface, pero aun es posible salvar 13 imágenes de 48K, o 4 de 128K en un disco.
SAVE 1;FRED S - salva la pantalla con nombre FRED en el disco 1.
Los usuarios del 128k pueden elegir entre sus dos pantallas posibles. Similarmente también se puede salvar el contenido de cualquier área de la memoria.

Load
sintaxis: L / LOAD []
donde:
Addr = la dirección de memoria donde se cargará y ejecutará
Carga la imagen "nombre" del drive o de la unidad por defecto como SAVE.

Erase
sintaxis: E / ERASE []
Borra el archivo del drive o unidad por defecto.

Move
sintaxis: M / MOVE
Este comando copia el archivo como usando uno o dos drives, siendo más lento el copiado con una sola unidad.

Alter
sintaxis: A / ALTER
donde:
addr = dirección de memoria a pokear
valor = el nuevo código a ingresar

Este comando permite la edición de los códigos del programa contenido en la memoria y así poder aplicar los POKES que uno encuentra en los magazines como "CRASH", sin tener que recurrir a trucos de carga o Multifaces.

Backup
sintaxis: B / BACKUP
Al igual que MOVE, este comando sirve para copiar datos entre discos, pero a diferencia del primero este copia todo el disco pasando los datos a otro . Esto incluye los espacios vacíos en disco. La única desventaja es que estos comandos son lentos. Con un solo drive habría que estar cambiando los discos continuamente debido a que lo hace copiando cada dato en 4 partes para evitar el perturbar la memoria principal del Spectrum. Al final resulta más rápido, cargar y salvar individualmente cada archivo.

Date
sintaxis: D / DATE
Este comando permite cambiar la fecha por defecto.

Keep
sintaxis: K / KEEP
Protege un archivo contra borrados accidentales con ERASE.

Unkeep
sintaxis: U / UNKEEP
Remueve la protección activada con el comando KEEP.

Quit
sintaxis: Q / QUIT
Sale del DOS y vuelve al punto desde donde se le llamó, o ejecuta la imagen previamente cargada con el comando LOAD.

Reset
sintaxis: *
Basta dar el comando "*" para resetear la computadora.

Swift Basic:
Carece de los comandos Alter, Backup, Date, Keep, Move y Unkeep propios del DOS, ni hay ningún equivalente Basic. Existen en cambio comandos exclusivos para el manejo de ficheros tanto secuenciales como directos.

Format
sintaxis1: FORMAT %;"nombre"[,max_fichas]
Permite formatear discos nuevos dejándolos listos para ser usados o reformatearlos eliminando todo su contenido previo.
Al igualque su versión del DOS, requiere que se le indique la cantidad máxima de fichas a contener.

sintaxis2: FORMAT %#; "T"/"B",baudrate,linelength
donde:
flujo = un valor de 0 a 15
baudrate = el rango de baudios
linelength = largo de las líneas a usar
T = modo texto
B = modo binario

Este comando formatea la toma serial para poder usar adecuadamente cualquier impresora que se le conecte a la interface.

Cat
sintaxis: Cat %[#flujo,] [,B/F]
CAT % es equivalente al comando CATALOGUE del DOS. Adicionalmente pueden usarse los canales del Spectrum, de modo que se puede enviar el catalogo mediante CAT %#3,0 a una impresora como la ZX, Alphacom o cualquiera otra conectada al interface (una vez formateado).

Save
Sintaxis: Save %;"nombre" [,Line/Code/Data/Screen$]
Sintaxis completamente similar a su versión de casete. Salva el programa dándole un "nombre" en el drive .

Load
sintaxis: Load %;"nombre" [,Line/Code/Data/Screen$]
Carga el programa "nombre" del drive .

Merge
sintaxis: Merge %;"nombre"
Igual que el comando de casete, carga solo programas basic del drive , con el "nombre" dado.

Erase
sintaxis: Erase %;"nombre"
Borra el programa "nombre" del drive .

Comandos de fichero:

Open#
Sintaxis: Open# %#;drive;"filename", <"R"/"W"/"A">[, "T"/"R",len]
donde:
las tres primeras opciones siguientes al nombre son:
R = read (abre el fichero para lectura)
W = write (abre fichero para escritura)
A = append (permite añadir nuevos datos al fichero antes creado)
y de las siguientes tres opciones:
T = test end of file (verifica la condición del fin de fichero, que normalmente da error)
R = random (permite trabajar con ficheros directos o aleatorios)
len = la longitud máxima de cada registro

Esta sentencia permite manejar tanto ficheros secuenciales como directos. Es posible abrir hasta cuatro ficheros a la vez, siempre y cuando se hallen todos en el mismo disco. El interprete Basic solo puede procesar ficheros a 0.5K por segundo.
Un fichero puede ser abierto al comienzo o al final, pero no puede ser rebobinado ni hay como moverse en un fichero de texto. Tampoco hay forma de descartar los datos del final sin crear un nuevo fichero.

Print#
sintaxis: Print #;datos
Envía la secuencia de datos a un fichero secuencial por un buffer mediante un flujo especificado. El canal debe haber sido abierto previamente para escritura "W" o añadir "A".

Input#

sintaxis: Input #;var;... [LINE] var$;...
Lee secuencialmente los datos de un flujo especificado y los pasa a una variable ya sea numérica o de cadena. El canal debe ser abierto previamente para lectura "R".

Inkey$#
sintaxis: [Let =] Inkey$ #
Lee el siguiente caracter de un fichero secuencial desde el flujo dado y leerá todos los caracteres del fichero, uno a uno.

%Eof
%EOF (end of a file) - es una función que permite detectar el final del fichero. Pero no hay modo de atrapar otros errores sin una utilidad tipo "ON ERROR".

Close#
Sintaxis: Close# %#
Cierra el flujo previamente abierto con OPEN#%.

Lprint y Llist
Una vez formateado el puerto serial con FORMAT es posible usar los comandos LPRINT y LLIST a cualquier velocidad con impresoras seriales.

In
Sintaxis: In %#,,
donde:
var = la variable usada para leer los datos del fichero directo
rec_num = el número de registro que sirve como puntero para acceder directamente a los datos

Out
Sintaxis: Out %#,,
Lo opuesto a IN%, esta instrucción permite escribir los datos directamente al fichero directamente por medio del puntero .
Con In% y Out% es posible escribir bases de datos hasta llenar el disco, eso son más de 600k de información pura.

Ejemplo de archivo directo:

```
1 REM ejemplo Swiftdisk II
5 CLOSE# %#4
10 OPEN# %#4;0;"print_list","R","R",5
15 REM cada registro ocupa 5 bytes
20 DIM p(1)
25 REM todos los datos son ingresados / leídos mediante matrices definidas con DIM
30 INPUT "Cual registro ?";x
40 IN %#4,p(1),x
45 REM dato en x pasado a variable p(1)
50 PRINT p(1)
60 GOTO 30
```

Emulación de Microdrive:

Basta con cargar un programa llamado EMUL para que el Swift se porte igual que un IF1. Al ejecutarlo, la rutina se almacena en la RAM

de la interface, permitiéndole al usuario correr programas comerciales para Microdrive sin modificarlos. Tanto el Swift-DOS y Swift-basic son deshabilitados y reemplazados por los comandos propios del MicroDRive.
El emulador es facil de usar. Toma 4 segundos cargarlo. Una vez hecho esto, FORMAT toma un área del disco y lo usa como si fuese un cartucho de Microdrive. Es posible crear hasta cuatro cartuchos virtuales en un disco y trabajar con todos ellos a la vez. Si bien el Swift-dos ve cada subdivisión como un archivo de 128k, desde el emulador CAT solo muestra hasta 50 archivos dentro de esa subdivisión. Opciones de CATalogo, ensamblado de archivos múltiples y otras operaciones "truqueras" trabajan muy bien. Entre los muchos programas probados con este sistema emulado están: Cheetah's Sound Sampler, Powerprint 2, Beta BASIC, HiSoft Pascal 1.6M, Oasis's Laser Genius, MIRA Pascal, Tasword I y Picturesque.

Es posible conectar un IF1 real y correr discos y cartuchos de MicroDRive a la vez, si bien no es posible usar los MicroDRives reales cuando se está en modo emulador, por razones obvias. En este caso, para intercambiar datos, Swift ofrecía un programa que pasaba los datos de cinta a un cartucho virtual automáticamente.