

INSTRUCTIONS

The Hackit interface allows any program to be stopped and hacked. The interface connects to the expansion port on the back of the computer and other peripherals can be attached to the Hackits own expansion port.

Before connecting Hackit, please ensure that the computer is switched off. The Hackit must be connected with the ribbon cable coming down from underneath the edge connector that fits the expansion bus on the computer.

The only two distinguishing features of the box itself are the Red reset button to the left of the box and the on/off switch at the right of the box.

The on/off switch is self explanatory, when switched on, every time the computer is reset by pressing the Hackits reset button, control will pass to Hackit. (Control will also pass to Hackit when the computer is reset by pressing the CONTROL-SHIFT & ESC keys together, or when the computer is turned on or finally when a CALL 0 or RST 0 instructions are executed by the Z80 Processor.)

The reset button will either completely reset the computer when the Hackit is switched off, or will stop the currently running program and pass control to Hackit if it is switched on.

When Hackit is switched on and the reset button is pressed, the following will happen:- The Z80 will reset, the Amstrad firmware (jumpblocks etc) will initialise. The screen will clear and the Amstrad power up message will show for a split second before the Hackit comes into action.

All memory from &40 to &A67F will remain intact for examination, alteration etc. The Hackit works solely in Hexadecimal, however it does have built in commands for hex to decimal conversion and vice-versa.

All commands must be entered at the ">" prompt and further prompts will allow parameters such as address, filenames etc to be entered. (in hex only).

Below is a list of commands and what they do.

DISC Initialises the disc rom (rom 7) allowing access to the disc system as opposed to the cassette system. When Hackit comes on, the disc system is not initialised in order for memory that the disc rom uses, can be examined, moved etc. The disc rom normally occupies memory from A67B to AB7F. The disc command allows you to specify at what address the disc rom will use.

EDIT This allows memory to be examined and altered. After specifying an address, the screen will fill with 188 bytes starting from the specified address. The cursor can be moved around the screen which will scroll up and down when the cursor reaches the top or bottom of the screen. The TAB key is used to toggle between hex & ascii editing whilst the ESC key will exit from the editor.

BASIC This command simply exits Hackit and reverts back to Basic and Amsdos. All memory will be cleared by this command.

CLS Clears the screen and moves the cursor to the first line on the screen.

CAT Catalogue a tape or disc depending on whether the tape or disc system is currently in operation.

RSX This command can be used to access any Roms that may be attached to the computer (i.e. Maxam, Utopia etc). This command will prompt you for the RSX name and will try and find and execute the relevant command. Please note that the relevant ROM must first be initialised.

Please see the Amstrad Firmware guide with reference to jumplock 154: KL INIT BACK at address BCCE in order to initialise a rom. This command can also be used for switching between tape and disc. i.e. !TAPE / !DISC

- PEN** Allows the foreground colour to be altered.
- PAPER** Allows the background colour to be altered.
- BORDER** Allows the border coloured to be changed.
- CALL** Allows a routine to be called. This command allows the individual Z80 registers to be loaded and displayed on return from the routine. Please note that because all the firmware jumpblocks are initialised, it is possible to use this command to access all the firmware routines.
- MEMORY** This useful command allows the whole user memory (0-C000) to be viewed on one screen. The display will show which blocks of memory are not occupied (blue areas) and which parts are occupied (yellow areas). Can be usefull for determining which areas of memory are actually used by the program and which are not. Patchy yellow/blue areas are often used for graphics data for games, whereas virtually solid yellow areas are usually program areas.
- HELP** Displays a list of available commands along with the version number of the Hackit in use.
- LOAD** Loads a file from tape/disc after the filename has been entered, an address can be specified as to where the file will load into. Pressing the return key at this prompt will load in the file at the address specified in its header.
- HEADER** Will display header information of the specified file on tape/disc.
- SAVE** Will save a specified area with a specified name onto tape or disc. Besides name, start address, length and entry address, and optional load address can be entered which will be the address that the file will normally load in at. It is therefore possible to save an area of memory from say 8000 that will normally load in at 4000. If you want the load address to be the same as the start address, simply press the return key when prompted for the load address. This option only works on files being saved to disc and not tape.
- COLOURS** Displays the 16 available inks and what they are set to. Can be usefull if a routine is called and on returning the inks need to be examined.
- MOVE** Moves a specified area of memory from one address to another. The original block of memory is filled with 0's as soon as it has been moved.
- COPY** Performs a similar job to move except then original block of memory remians intact.
- FILL** Fills a block of memory with a specified value. Can be used to clear areas of memory.
- DISASSEMBLE** Disassembles the relevant area of memory into Z80 mnemonics. Besides a start and end address, if a filename is given, an ascii file will be created on tape or disc containing the disassembly. This can then be loaded into an assembler for re-assembling.
- HEXTODEC** Will convert 16 bit hexadecimal (0-FFFF) numbers to decimal.
- DECTOHEX** Will convert decimal numbers in the range 0-65535 to

hexadecimal.

SEARCH Will search for an ascii or hex string in memory. Any addresses that contain occurrences of the string will be displayed. At least one occurrence will always show, this is because the original string must be stored in memory for comparison, and the search will obviously find this string.

PRON Printer On. This command will print onto your printer any data that Hackit will display on the screen. The only exception to this is when using the EDIT command. However, pressing the CONTROL and COPY keys together at anytime will perform a screen dump.

PROFF Turns off the above mentioned printer on command.

BANK Will allow access to the 2nd 64K bank on an 6128 or expanded 464/664. Only 16K blocks can be paged in at address 4000-8000 and are numbered 1-4. Up to 16 of these 16K blocks can be accessed if a 256k memory expansion is fitted.

ALTERNATE Please note that this command will only work on a 464/664 with a DK Tronics 64K memory expansion. It will not function correctly on a 6128 or standard 464/664. On an expanded 464/664, it is possible to get a program running in the second bank of memory and let the computer think it is running in the normal bank of RAM. Once the program is running, it is possible to enter Hackit (by pressing the reset button) and the whole of the 2nd 64K bank will still be intact. This includes the screen memory etc. After typing in the ALTERNATE command, the cursor will disappear. At this stage turn off the HACKIT and press the CONTROL-SHIFT and ESCAPE keys together. Anything displayed on the screen will become jumbled as the mode is changed. At this point you must type in the load instructions to load in the game. You will not be able to see what you typing, but the computer will still be accepting the information. Once the game has loaded, turn on the Hackit and press the reset button to access the hackit. if you now use the BANK command to select bank 4 and then do a COPY command to copy from 4000 bytes from address 4000 to address C000 then the screen from the game will be copied onto the current screen. The rest of the games code can be edited by examining banks 1,2 and 3 of the 2nd 64K of ram.

OUT Outputs a byte to a port.

IN Reads in a byte from a port.

CLEAR Fills memory from 40 to AB80 and all extra banks of memory with 0.

RESTORE Restores the colours to black on white. Usefull if you call a routine that sets the colours to an illegible combination.

POKE As the basic POKE command

PEEK As the basic PEEK command.

ASSEM The assembler is an in-line Z80 assembler and assembles a line of code when it is entered. It is only a basic assembler and does not support labels. All instructions must be entered in UPPER CASE and all numbers must be preceded by a & sign. All numbers must be signed hex integers. To quit from the assembler, simply enter Q as an instruction. Example instructions:-

CALL &3200	BIT &1,L	LD A,(IX&28)	DEC (HL)	JR &34
JP &5000	OR &78	INC DE	RES &6,A	DJNZ &56

Obviously in order to make the most of HACKIT it is important to have a working knowledge of machine code. Hackit has not been developed in order to facilitate automatic tape to disc transfer, it has been developed to enable users to stop

and hack programs. Obviously with the ability to save areas of memory, it may be possible to copy programs, however this is not the prime reason for designing Hackit.

Some programs will not work with Hackit turned on. To remedy this, load the program with Hackit turned off, and turn it on when the program has loaded. Because any program can be stopped at any time, even loaders can be stopped to be disassembled to see how they work.

We hope that you enjoy using Hackit.