# VIC KIT 4
# USERS GUIDE

STACK

COMPUTER
SERVICES LTD

## INTRODUCTION.

VICKIT 4 is a 4k EPROM containing firmware to allow the user to load and save on cassette up to 6 times faster. VICKIT 4 also features a machine code monitor, Hexadecimal calculator and link commands to use powerful VICKIT 5 assembler firmware.

VICKIT 4 is designed to run with the programmers aid chip VICKIT I or the hi-res utility chips VICKIT II/III.

As VICKIT 4 is only a chip it requires a board to link it to the Vic-20. This can be either a STACK ROM Switchboard which allows up to four 4k EPROMS on a single VIC or a STACK SUPERCHARGER which allows up to two 4k EPROMS on a Vic-20. It is possible that other boards may take extra ROM/EPROMS. These boards should have a 4k,2532 compatible socket addressed $ AØØØ to $ AFFF.

## FITTING VICKIT 4.

a). To Stack Rom Switchboard.

Remove the covers of the Rom Switchboard by pressing out the plastic rivots. This will reveal four 24 pin sockets. VICKIT 4 is designed to fit into either IC7 or IC5. If the RAM area of the Vic is fully populated use socket IC5. If not then IC7 should be used.

Locate the notched end of the VICKIT 4. Orientate the VICKIT 4 over the required socket with the notch closest to the gold fingers. Now gently push the chip home into the socket, making sure every pin sits correctly.

The board can now be plugged into the VIC. Arrow will auto start. If the chip does not respond try the change over poke and warm start with SYS40999.

When you are happy with the installation replace the covers on the Rom Switchboard.

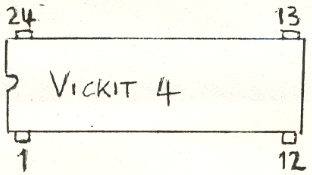
b). To a Supercharger.

Again remove the plastic rivots and covers from supercharger. Located inside are two 24 pin sockets VICKIT 4 ahould be fitted into the socket furthest away from the gold fingers i.e. B A 2. The notch on the EPROM should be closest to the two intergrated circuits IC1 and IC2.

Replace covers and plug into the Vic memory expansion. VICKIT 4 will then auto start on power up.

c). To other type boards.

First check with the manufacturer that the socket is 4k, 2532 compatible and addressed 40960 to 45056 or Hexadecimal A000 to Afff. Next locate pin 1 of the socket.
To locate pin 1 of the VICKIT place the VICKIT on a table resting on its 24 pins, orientate the VICKIT so that the notch is on left hand side (TOP VIEW). Pin 1 is the pin nearest the viewer on the left most side. (BELOW THE NOTCH).



NOTE POWER SHOULD ALWAYS BE OFF WHEN PLUGGING ANY DEVICE IN OR OUT.

## HIGH SPEED CASSETTE OPERATION.

The four commands are ←S ←L ←V and ←A are used to SAVE, LOAD, VERIFY or APPEND at high speed (3600 baud). All except the shortest of programs will be handled 6 to 7 times faster than the standard speed.

← is used to indicate the key at the top left corner of the VIC Keyboard.

Program names are limited to 16 characters.

Except for a program save operation, the program name is optional and if the name specified is shorter than the name used when the program was saved, a match on the beginning of the file name(corresponding yo your keyed input) is sufficient.

After each cassette Input/Output operation, the length of the program handled will be displayed on the screen.

←S"PROGRAM NAME" is used to save a BASIC program on the Datasette with a 4 second leader preceeding the program. When using this command, the tape should be positioned on the magnetic portion of the tape and not on a long non-magnetic leader.

←T"PROGRAM NAME" is used to save a BASIC program on the Datasette with a 10 second leader preceeding the program.

←A"PROGRAM NAME" is used to add a BASIC program from the Datasette to the BASIC program in the memory of the VIC-20. The append function works only for BASIC programs in the same manner as the TOOLKIT, that is the program appended will be added to the end of the existing program in memory without regard to line numbers.

←V"PROGRAM NAME" is used to verify that the BASIC program on you Datasette matches the program in memory. In the event that it does not, ?? will be displayed. Note that this is also a method of skipping a program on the Datasette for example to save another program after a particular program on tape. Note that ←V will always display ?? when used after ←A.

←S" PROGRAM NAME",SSSS,EEEE (SSSS=start address and EEEE=end address plus 1) is used to save a machine language program or block of memory with a 4 second leader. The ←T form of this instruction may be used to save with a 10 second leader.

To load or verify a machine language program, use SHIFT of the L and V keys.

## TAPE POSITIONING.

The command ←P where d is digit from 1 to 9 is used to advance the cassette to one of nine positions using the Fast Forward button. Each block skipped is sufficient for a 16k program.

## HEXADECIMAL CALCULATOR/CONVERTOR.

You may enter this mode by typing ←H which will cause two counters to appear on the bottom line of the screen. At first they will both be zero, but if you type in hexadecimal number you will see it appear in the right-hand counter while its decimal equivalent is displayed in the left-hand counter.

You can add or subtract using the plus, minus and equals keys and you can switch from hexadecimal to decimal input or back again by using the* key. The H or D displayed at the end of the bottom line indicates which form of input, hexadecimal or decimal is currently in use.

Repetition of the equals key causes the last addition or subtraction to be repeated.

For multiple conversions from decimal to hexadecimal or vice versa, enter the equals sign after keying in the number to be converted. Remember that the entry mode is indicted at the end of the bottom line.

The largest decimal value that can be handled is 65,535 with the hexadecimal maximum at $FFFF.

To exit from this mode type X.

## DEACTIVATION.

If, for some particular reason, you need to disable the ARROW functions, ← O(oh,not zero) may be used. You can reactivate by keying in a SYS40997.

## TECHNICAL NOTES.

It is possible to write/read blocks of data from within a machine code or BASIC program. Use JSR $A855 (SYS 43093) to write a block to cassette. Start and end address should be storedin lo-hi form locations $03F4 to $03F7. The file name of up to 16 characters is stored at $03E4 to $03F3.

To read a block from cassette, use JSR $A85B (SYS 43099). Blocks of any length may be handled in this manner provided that the start address is above $03FF.

## TEXT EDITING.

There is a FIND command with an optional replacement field were both the search string and the replacement string may have up to 20 characters each. The format is:-

← F" search string"/replacement string/

If you want to search and replace with nothing (delete the search string), use the left arrow key (← ) as the replacement string.

If there is no replacement string, each line where a match is made will be listed and the search will continue. This may be slowed down by the CTRL key or stopped by using the STOP key.

If you have specified a replacement string, a match will result in the old line being displayed with the new line repeated just below the old line with the replacement made, including memory. If you do not desire this replacement, you simply use the cursor Up key to get the old line and press return. If the replacement was correct, nothing has to be done.

← N first line number, increment may be used to initiate automatic line numbering. To exit from AUTO type ← Q.

← R first line number, increment may be used to renumber the lines in an assembler source program being prepared or modified for the assembler. This is not intended for BASIC programs and therefore does not take into account GOTO,GOSUB, THEN and ON N GOTO statements. The default values are first line number 10withan increment of 10.

← D first line number, last line number may be used to delete block of lines.

← E initiates the execution of the assembler. (VICKIT 5 required).

← B initiates execution of EZASM at the starting address plus 5 when user coded modification of the output character stream is used. (VICKIT 5 required).

← M address, end address, new start address may be used to move block of memory. The original block is not modified unless there is a conflict between the starting address and the new starting address. Note that code is merely moved and logically relocated. Immediately after each byte is moved, the byte is compared to memory and if there is a difference due to a move ROM or a bad RAM address, the contents and the address will be displayed on the screen.

Continued..........

←C beginnig address, end address, second beginning address may be used to compare any two blocks of memory. Any differences will be displayed on CRT.

←X may be used to go onto the machine code monitor. The commands now at your disposition are:-

M BBBB EEEE (beginnig and end addresses in hex) to display a block of memory, 4 bytes per screen line. If EEEE is not specified, 4 bytes will be displayed. Moving the cursor and changing displayed bytes then keying a return will result in a change of the contents of memory.

←G BBBB will result in a jump to the specified address. If no address is specified, the contents of the PC will be used.

R results in a register display. The zones are defined as follows PC is the program counter, SR is the status register, AC is the accumulator , XR and YR are the registers X and Y and SP is the stack pointer. Changing these fields (always followed by a return) will result in the new values being used upon a G instruction. When a BRK instruction (hex 00) is inserted in a machine language program, the monitor will be entered with the current values of all the above registers displayed.

S" file name",01,BBBB,EEEE (beginning address and end address plus one) may be used to write a block of memory in the standard (slow non-Arrow) format.

L" file name", may be used to laod a block of memory saved via the 5 command above.

X may be used so return to normal operation (READY message).

## DISCLAIMER.

Whilst every effort has been made to provide a flexible , reliable and above all low-cost product STACK COMPUTER SERVICES LTD. wish to point out that no claim is made for complete compatibility with any other equipment or program. The information given is believed to be accurate but no liability can be accepted for the consequences of any error. Ours is a policy of continued development and we therefore reserve the right to alter the design of sppecifications without prior notice.

## REQEST FOR INFORMATION.

In order to provide the user with as much support as is practical, we would appreciate if any useful comments or hints could be forwarded in to writing to:-

PRODUCT DEVELOPMENT
STACK COMPUTER SERVICES LTD
290/298 DERBY ROAD
BOOTLE,
LIVERPOOL L20 8LN.