

MULTI ROM CARRIER

DISTRIBUTED BY
NEEDMORE DISTRIBUTION
28 FARRIERS WAY INDUSTRIAL ESTATE
NETHERTON MERSEYSIDE
L30 4XL

051 521 2202

COMPUTER SERVICES LTD
N.

INDEX

<u>Page</u>	<u>Description</u>
1)	Programming the Multi ROM Carrier.
2)	The Vickit Series.
3)	Placement of Vickits.
4)	Hints.
5)	Auto-start EPROM/ROM Firmware.
6)	Example using Vickits 1, 3 and 4.
8)	Example using Vickits 1, 3 and 4 with FULLY EXPANDED MEMORY.
9)	Examples using Vickits 4 & 5.
10)	Disclaimer & Request for information.

MULTI ROM CARRIER.

DESCRIPTION.

The Stack Multi ROM Carrier allows up to four 4K ROMs to share one 8K block on the VIC. The board allows two 8K blocks to share the same address space by having only one block enabled at a time.

A1 and B1 share the slot A000-AFFF (40960-45055) and A2 and B2 share the slot B000-BFFF (45056-49151). Hence one 8K block is A1 and A2 and the other 8K block is B1 and B2. Switching from block to block is achieved from software, utilising the top two bytes of user memory. Reading or writing to either of these two addresses will select the corresponding 8K block.

PROGRAMMING.

To alter the block selected, the following procedure must be performed:-

To select B1 and B2	POKE 32766,1
To select A1 and A2	POKE 32767,1

USE WITH 24K and 24K + 3K MEMORY EXPANSION.

For users with a 24K or 24K + 3K memory expansion the VIC initialises with A1 and A2 active. The use of the top two memory locations in the memory expansion means that with a 24K or 24K + 3K memory expansion, the top of memory pointers must be altered. This is done by entering:-

POKE 52,127: POKE 56,127.

(This is required if a large programme is present which might right into the two bytes concerned.)

after power up.

The required block should then be selected as above.

Note:- Top of memory is 32768 if +24K is fitted as extra RAM.

Location 55 and 56 are Top of memory pointer

to find top of memory Print Peek (55) and Print Peek (56).

Top of memory is $\text{Peek}(56) * 256 + \text{Peek}(55)$ (or $128 * 256 + 0$)

Similarly for locations 52 and 51 (string variable pointer).

By poking location 52 and 56 with 127 we have changed the top of memory pointer to $127 * 256 = 32512$. 'ie memory reduced' by $32768 - 32512 = 256$ bytes. (256 bytes = 1 page in Vic terminology).

Why they have to reduce memory by 256 bytes as only 2 bytes are used ie 32767 and 32766 I do not know? ie Poke 56,127: Poke 55,254.

THE VICKIT SERIES:-

The VICKIT Series comprises of 5 EPROMS, each adding new basic commands to your VIC computer.

VICKIT 1 :-

✓ The original VICKIT is the first of a series of EPROMS developed for the VIC-20.

When the VICKIT has been installed and enabled, 9 extra commands are added to BASIC these are:-

AUTO, DELETE, DUMP, FIND, HELP, OFF, RENUMBER, STEP, and TRACE.

VICKIT 2 :-

A VICKIT 2 is a 4k ROM which adds 21 extra commands and statements to BASIC.

9 programming aid commands are:-

AUTO, DELETE, DUMP, FIND, HELP, OFF, RENUMBER, STEP, TRACE.
12 Hi-res graphics statements.

TEXT, GRAPHICS, LINE, CLEAR, DRAW, PUT, FILL, SET, POINT, CIRCLE, RESET, INVERT.

VICKIT 3 :-

VICKIT 3 is a 4k EPROM addressed to \$ A000 produced by STACK for VIC-20 programmers.

It provides 20 Hi-res graphics commands:-

SCREEN, BORDER, GROUND, COPY, WRITE, FONT, OFF, SWITCH, TEXT, GRAPHICS, LINE, CLEAR,
DRAW, PUT, FILL, SET, POINT, CIRCLE, RESET and INVERT.

VICKIT 4 :-

VICKIT 4 provides the VIC user with a machine code monitor and improved cassette performance on a 4K ROM addressed \$ A000. Cassette functions can be executed at up to six times normal speed for all but the smallest programs. Special features include source text editor and hexadecimal calculator.

VICKIT 5 :-

The VICKIT 5 provides a superior 6502 Assembler, allowing the user to create and run programs written in machine code. (Must be used with Vickit 4).

PLACEMENT OF VICKITS:-

Up to four of these utility chips can be added to the MULTI ROM CARRIER. The number and choice of VICKITS added gives rise to different siting in the Rom Carrier. Below are some suggested combinations:-

	<u><24K or (<24K + 3K)</u>	<u>24K or (24K + 3K)</u>
<u>VICKIT 1</u>	VICKIT 1 in B2	VICKIT 1 in A2
<u>VICKIT 1 + 3</u>	VICKIT 1 in B2 VICKIT 3 in B1	VICKIT 1 in A2 VICKIT 3 in A1
<u>VICKIT 1 + 4</u>	VICKIT 1 in B2 VICKIT 4 in B1	VICKIT 1 in A2 VICKIT 4 in A1
<u>VICKIT 1 + 4 + 5</u>	VICKIT 1 in A2 VICKIT 4 in B1 VICKIT 5 in B2	VICKIT 1 in B2 VICKIT 4 in A1 VICKIT 5 in A2
<u>VICKIT 3</u>	VICKIT 3 in B1	VICKIT 3 in A1
<u>VICKIT 3 + 4</u>	VICKIT 3 in A1 VICKIT 4 in B1	VICKIT 3 in B1 VICKIT 4 in A1
<u>VICKIT 3 + 4 + 5</u>	VICKIT 3 in A1 VICKIT 4 in B1 VICKIT 5 in B2	VICKIT 3 in B1 VICKIT 4 in A1 VICKIT 5 in A2
<u>VICKIT 2</u>	VICKIT 2 in B2	VICKIT 2 in A2
<u>VICKIT 2 + 4</u>	VICKIT 2 in B2 VICKIT 4 in B1	VICKIT 2 in A2 VICKIT 4 in A1
<u>VICKIT 2 + 4 + 5</u>	VICKIT 2 in A2 VICKIT 4 in B1 VICKIT 5 in B2	VICKIT 2 in B2 VICKIT 4 in A1 VICKIT 5 in A2
<u>VICKIT 4</u>	VICKIT 4 in B1	VICKIT 4 in A1
<u>VICKIT 4 + 5</u>	VICKIT 4 in B1 VICKIT 5 in B2	VICKIT 4 in A1 VICKIT 5 in A2
<u>VICKIT 1 + 3 + 4 + 5</u>	VICKIT 1 in A2 VICKIT 3 in A1 VICKIT 4 in B1 VICKIT 5 in B2	VICKIT 1 in B2 VICKIT 3 in B1 VICKIT 4 in A1 VICKIT 5 in A2

If a Vic has 24K + 3K RAM expansion then the sockets which are on line after power up are:-

A2 address \$ B000, decimal 45056

A1 address \$ A000, decimal 40960

CONTINUED.....

If the Vic has less RAM expansion then the sockets on line at power up will be:-

B2 address \$ B000, address 45056

B1 address \$ A000, address 40960

HINTS.

- 1) In general when any EPROM/ROM is initialized (SYSCALL) it alters page 0 registers within the Vic. It may not conform with the new EPROM/ROM on line. Hence the user should always use the disable facility of the EPROM/ROM before switching.
 - 2) WARM and COLD start. A COLD start usually sets up a system and destroys any program in memory. WARM start usually calls the initialization of EPROM/ROM without loss of the program in memory. This should be used after switching ROM banks if the program is not to be corrupted.
 - 3) It is suggested that the user exercises the utilities of the system and becomes familiar with its limitations and his own programming ability before attempting long programs (i.e. Write a short program and test it).
-

AUTO START EPROM/ROM FIRMWARE

When Vic powers up, it first checks whether an auto-start ROM is present and if so, it initialises it. It then calculates the amount of RAM which is available for BASIC and if this is 24K, it 'switches in' the appropriate bank of EPROM/ROM. If this is different to the bank in which the auto-start EPROM/ROM is placed, then a crash may occur.

i.e. 24K memory : VICKIT 1 Socket B2

: VICKIT 3 Socket B1

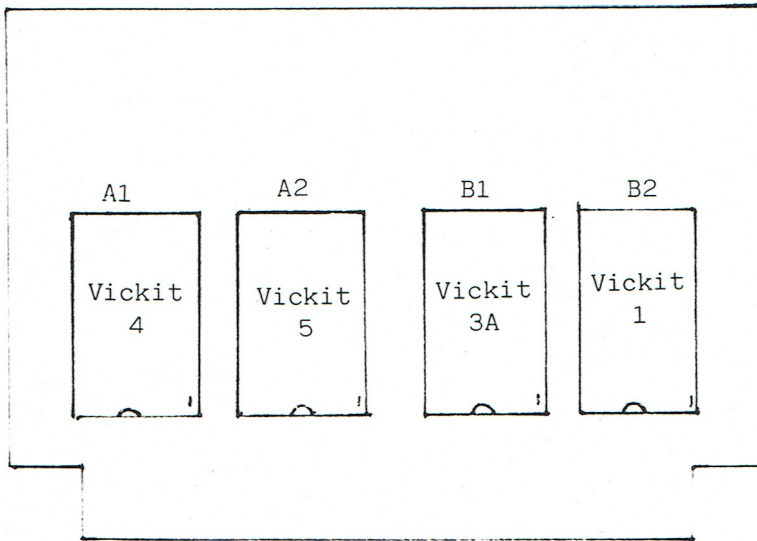
Auto Start : VICKIT 4 Socket A1

VICKIT 4 is an auto start ROM. Hence one would expect it to be on line at power up. But careful study shows why it is not.

It is advisable to follow sequence below:

1. Power up.
 2. Drop Memory Pointer.
 3. Select Bank.
 - 4A. Warm Start, Auto Start EPROM/ROM (Arrow).
 - 4B. SYS CALL (Cold start). Non Auto Start EPROM/ROM.
-

Example using Vickits 1, 3 and 4.



The operations shown on pages 6 to 8 require the Vickits to be positioned as shown (left) and not necessarily as per the table on page 3.

Please remember that the Graphics command of the Vickit 3 requires that the 3K RAM area \$0400 - \$0FFF(1024-4095) is filled. On pages 6 to 8, the term 'unexpanded' is used to represent a VIC plus 3K RAM expansion but no further memory expansion.

32767,1

32766,1

BLK-A

BLK-B

To write a program using VICKIT 1 and 3 with the high speed save.

(The sequence of operations shown on pages 6 & 7 are for an unexpanded Vic).

SYS 4096 x 11

Powers up VICKIT I

SYS 4096 x 10

Powers up Vickit 3

AUTO

10 Graphics

20 Clear 3

30 For L = 10 to 40 STEP 3

40 Circle (80, 80), L

50 Fill (80, 80)

60 NEXT

70 Fill (140,140)

80 For Y = 1 to 500 : NEXT

90 Text Upper

Program

RUN

SYS 58459

Disables Vickits 1 & 3.

Poke 32767,1

Switches to bank A.

SYS 40997

Warm start Arrow

←S" TEST

Saves Program

Continued.....

← 0

POKE 32766,1

SYS 4096 x 11

SYS 40963

RUN

LIST

Disable Arrow

Switches to bank B.

Power up VICKIT I

Warm start Vickit 3.

Example using vickits 1, 3 and 4 with FULLY EXPANDED MEMORY.

POKE 52,127 : POKE 56, 127
POKE 32766,1

SYS 4096 x 11
SYS 40960

10 Graphics
20 Clear 3
30 For L = 10 to 40 STEP 3
40 Circle (80,80), L
50 Fill (80.80)
60 NEXT
70 Fill (140,140)
80 For Y = 1 to 500 :NEXT
90 Text Upper

RUN

SYS 58459

POKE 32767,1

SYS 40997

← S

← 0

Power Off

Power Up

POKE 52, 127 :POKE 56, 127

POKE 32767,1

SYS 40997

← L

← 0

Poke 32766,1

SYS 4096 x 11

SYS 4096 x 10

RUN

Lower memory pointer

Select bank B (Vickits 1&3)

Power up VICKIT 1

Power Up VICKIT 3

Program

Disable both VICKIT 1 and 3.

Select bank A(Vickits 4&5).

Warm Start Arrow

SAVE PROGRAM

Disable Arrow

Lower Memory Pointer

Select bank A(Vickits 4&5).

Warm Start Arrow

LOAD

Disable Arrow

Select bank B(Vickits 1&3A)

Power up Vickit 1

Power up Vickit 3

Note; For an unexpanded VIC this program should still work due to using the bank select poke at the beginning of the program.

Select Vickits 4 & 5 if not already selected on power-up (as explained elsewhere in manual).

(a) For Unexpanded VIC or VIC + 3K RAM expansion:-

10 .0 4596

20 PRINT = \$E742

30 LDX 0

40 LOOP TXA

45 JSR PRINT

50 DEX

60 BNE LOOP

70 RTS.

Press

← E

When B?

Type 1

if no errors

Type Y

SYS 4596

This prints out the character set.

Stop with RUNSTOP/RESTORE.

(b) For VIC with greater than 3K expansion:-

If the Vic contains extra memory move the origin to 8194

ie LINE 10 .0 8194 (Oh. Not zero)

and to start the program

SYS 8194

Use RUN STOP Restore to END PROGRAM

DISCLAIMER

Whilst every effort has been made to provide a flexible, reliable and above all low-cost product; Stack Computer Services Ltd. wishes to point out that no claim is made for complete compatibility with any other equipment or programs. The information given is believed to be accurate but no liability can be accepted for the consequences of any errors. A policy of continuing development is employed and the company therefore reserves the right to alter specifications without prior notice.

REQUEST FOR INFORMATION

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

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

(C) 1983 Stack Computer Services Ltd.

Rom slot installed and power up.

Print Peek (43) = 128

" " (44) = 18

as expected bottom of memory unaffected.

Print Peek (55) = 0

Print Peek (56) = 128

As before ?

Print Peek (52) = 128

" " (51) = 0

Print Peek (32767) = 255

" " (32766) = 150



MOTOR SHOW OCT 18 → 26 th

Entry £3-00

Trade Days - Oct 15-17 9am → 7pm

Public Days - Oct 18-25 9.30 → 7pm

Vic 20

No Rom socket installed:-

28159 Bytes Free

Print ~~Poke~~ (43) = 1

Print ~~Poke~~ (44) = 18

ie $18 * 256 + 1 = 4609$. ✓ Start of Basic
ie Bottom of mem

Print ~~Poke~~ (55) = 0

Print ~~Poke~~ (56) = 128

ie $128 * 256 + 0 = 32768$. End of Basic
ie Top of mem.

Print ~~Poke~~ (52) = 128

" " (51) = 0

1 page = 256 locations

So why not { Poke 56, 127: Poke 55, 254
 { Poke 52, 127: Poke 51, 254

if only two bytes are required for Eprom.

LABELS REQUIRED

VICKIT 1 - ON SYS 450(56)
 OFF SYS 584(59) " "



0 = (22)
 851 = (52)
 ?

0 = (22)
 851 = (52)
 ?

0 = (22)
 851 = (52)
 ?

0 = (22)
 851 = (52)
 ?

0 = (22)
 851 = (52)
 ?



At 22 - 81T20 WCH2 20TOM

00-23-00

mgf & map 71-21-150 - 20TOM

mgf & map 71-21-150 - 20TOM