Final Expansion 3 User Guide

Version 1.8

Copyright Information

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License, for details see

http://creativecommons.org/licenses/by-sa/4.0/

Any trademarks mentioned are the property of their owners.

Version Information

Version	Date	Changes
1.8	2022-01-07	Corrected RAM/ROM mode write behaviour
1.7	2021-12-18	Noted wedge save command is not in mini-wedge
1.6	2021-12-02	Corrected mode register information
1.5	2021-08-20	Added example setting RTC, expanded directory examples
1.4	2018-03-18	Fix references and section levels
1.3	2017-09-09	Added DIP switch settings
1.2	2017-05-01	Added container files
1.1	2017-03-12	Minor corrections
1.0	2017-03-08	First revision

Authors

Simon Rowe <srowe@mose.org.uk>

Contents

Introduction

This guide describes the features of the Final Expansion 3 cartridge for the VIC-20 and how to use them. It also includes reference material for programmers to write software that uses the advanced capabilities of the cartridge.

The information given in this guide is believed to be accurate, however because there are many hardware and software variations the authors cannot guarantee that a specific system will function exactly as described.

The authors would be grateful to hear from you if you find a mistake or omission in this guide.

Where to Get Help

As there are several versions of the cartridge produced by different manufacturers the best source of information specific to yours is wherever you purchased it from.

The on-line forum Denial, http://sleepingelephant.com/denial/, is a community of VIC-20 users. They may be able to provide information and suggestions but cannot be expected to resolve your specific problems.

Cartridge Overview

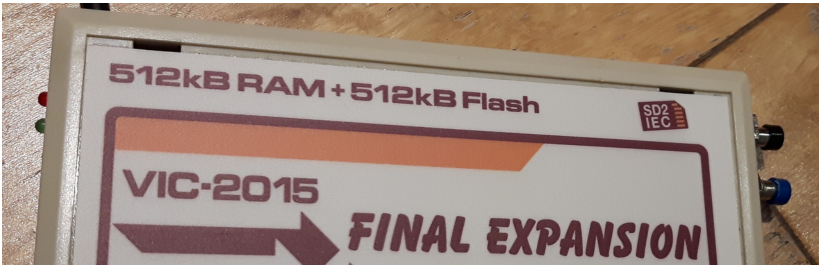
The Final Expansion 3 (FE3) cartridge expands the VIC-20 with:

- 512KB of Random Access Memory (RAM)
- 512KB of Electrically Erasable Read Only Memory (EEPROM)
- storage to Secure Digital cards emulating a serial bus disk drive (SD2IEC)

The memory expansion is highly configurable, both from interactive menus and by commands contained in user-created files. Virtually any type of memory map can be set up to allow all manner of programs to be run.

Additional commands are available to make disk devices easier to use, these are provided using a software wedge (which can be disabled) similar to those that work with physical disk drives such as the 1541.

Loading and saving to serial bus devices is accelerated if they contain support for JiffyDOS, this includes the integrated SD2IEC device.



The two indicators which are located on the left of the cartridge relate to the SD2IEC device

- error (top, red) – flashes when an error message is present on the command channel
- activity (bottom, green) – lit when a file is open on the device

The two push buttons on the right of the cartridge provide different forms of system reset

- VIC-20 system reset (top) – system resets but memory configurations etc. are retained
- VIC-20 and Final Expansion 3 reset (bottom) – system resets with defaults identical to those at power on.

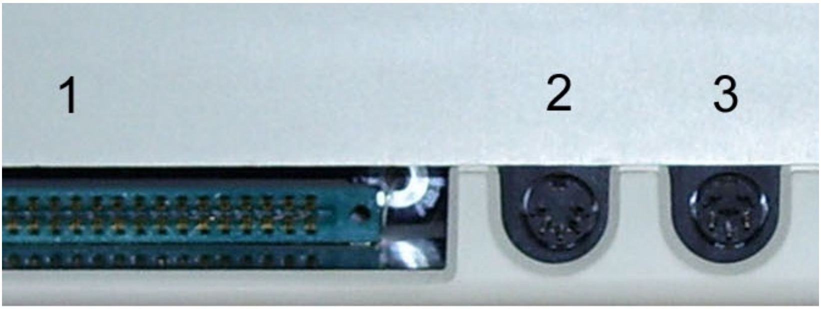
Quick Start

This section aims to help the first time user get up and running as quickly as possible. It also covers likely issues that the user may run into.

Setup

The cartridge should be connected to the VIC-20 with the power switched off. It may either be inserted directly into the cartridge port or into a slot of a cartridge expander.

The SD2IEC should then be connected to the serial port of the VIC-20. The serial port is the rightmost round socket (3) when viewing the rear of the VIC-20



Some FE3 cartridges have an integral lead with a plug, others have two sockets. Either

- Connect the plug to the serial port
- Connect one plug of a serial cable into the serial port and one into either socket on the FE3

An SD card should be prepared on a modern PC by formatting it. Only SD cards that are FAT formatted can be used with the SD2IEC device.

Insert the SD card, face down, into the slot on the top of the FE3, it should click into place. The card may be removed by gently pressing the top of the card. MicroSD cards may be inserted if an adapter is used.

After checking the cartridge is fully inserted and the serial cable is attached the VIC-20 can now be powered on.

Common Problems

No startup menu or extra memory but the SD2IEC works correctly

Your cartridge is missing its firmware. See “Appendix B – Updating Firmware” for how it to flash the cartridge.

Startup menus appear, memory present but SD2IEC device is not present

Check the cabling between the FE3 and the VIC-20.

Check the DIP switches, see “Appendix A – DIP Switches”.

SD2IEC device is present but it is unable to read or write to the SD card

Check that the SD card is formatted with a FAT file system. Cards larger than 32GB may not be compatible, try a smaller capacity card.

Start Up Menus

A sequence of menu screens can be used to easily access the features of the FE3.

Main Menu

When the VIC-20 is powered on a menu with the following options is presented

- Select common memory configurations
- Load programs from SD2IEC
- Load programs from flash
- Store programs to flash
- Change the SD2IEC device number

Options are selected by pressing the function keys (along with the Shift key if necessary).

Pressing the **F8** key returns from any submenu.

Bypassing the main start up menu can be achieved by holding down one of the following keys during power on:

- **SHIFT** – wedge enabled, no additional memory
- **C=** – wedge disabled, no additional memory
- **CTRL** – enter “Disk Loader” menu (see below)

The common configuration of all memory blocks being filled is available by pressing **F7** (with the command wedge) or **F8** (without the command wedge).

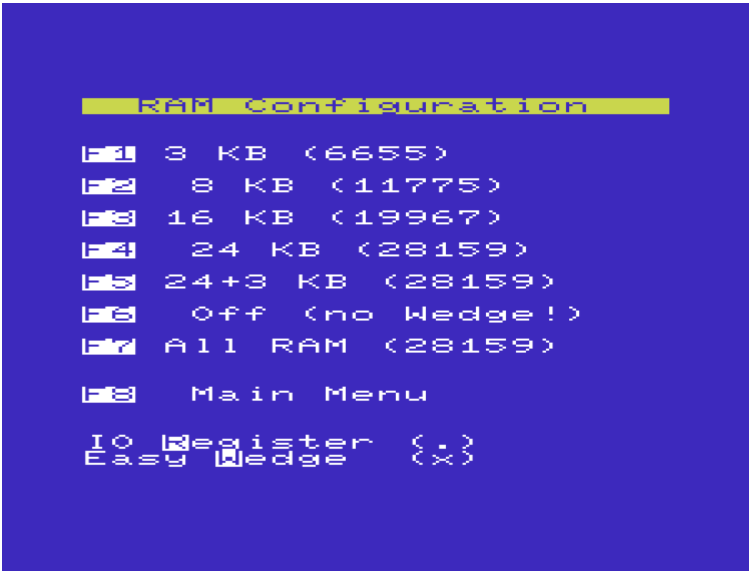
The device number used by the SD2IEC device can be changed by pressing the **+** and **-** keys. Values between 8 and 15 are supported.

Pressing the **F4** key displays a brief description of the commands the wedge provides. See “Wedge Commands” on page 23 for more details.

Pressing the **C** key displays a list of credits.

RAM Manager

This submenu allows different memory configurations to be set up. After selection the system will enter BASIC as normal.



```
   RAM Configuration

F1 3 KB (6655)
F2  8 KB (11775)
F3 16 KB (19967)
F4  24 KB (28159)
F5 24+3 KB (28159)
F6  Off (no Wedge!)
F7 All RAM (28159)

F8  Main Menu

 IO Register (.)
Easy Wedge   (x)
```

The configurations available are summarized in the following table:

Key	Memory	Blocks	Wedge?
F1	3K	RAM1,2,3	Yes
F2	8K	BLK1	Yes
F3	16K	BLK1,2	Yes

Key	Memory	Blocks	Wedge?
F4	24K	BLK1,2,3	Yes
F5	3K + 24K	RAM1,2,3 & BLK1,2,3	Yes
F6	-	-	No
F7	3K + 24K+8K	RAM1,2,3 & BLK1,2,3,5	Yes

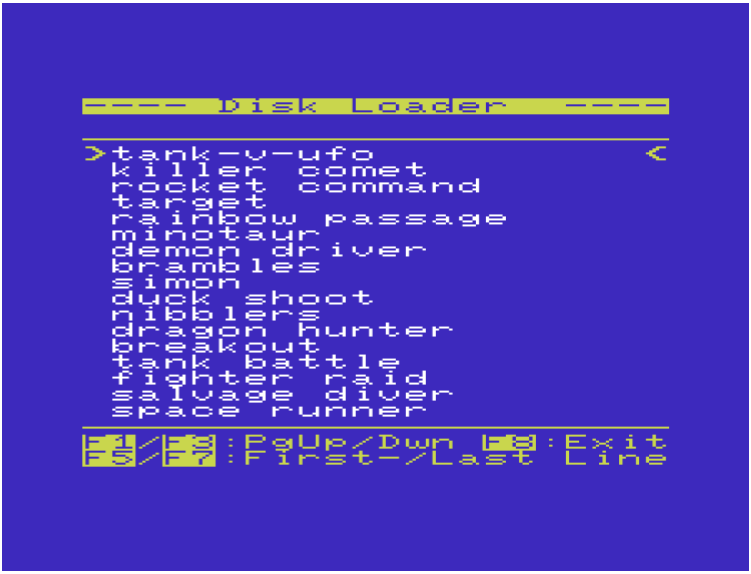
At the bottom of the menu two checkboxes are shown:

- I/O registers (**R**) - Allow software access to the registers which control the memory configuration of the cartridge
- Command wedge (**W**) – Enable commands listed on page 23

Pressing the given key toggles each checkbox.

Disk Loader

This submenu allows memory configurations to be set up and program files to be loaded from the SD card by selecting an entry from a list. The selections are defined in loader files, see “Loader Files” on page 29 for details of how to create them.



```
----  Disk  Loader   -----

>tank-v-ufo              <
 killer comet
 rocket command
 target
 rainbow passage
 minotaur
 demon driver
 brambles
 simon
 duck shoot
 nibblers
 dragon hunter
 breakout
 tank battle
 fighter raid
 salvage diver
 space runner

F1/F3:PgUp/Dwn  F8:Exit
F5/F7:First/Last  Line
```

The menu is navigated by using either the cursor keys or the joystick, an entry is selected by pressing **RETURN** or the fire button. When there is more than one page of entries the **F1** and **F3** keys can be used to move between pages. The **F5** and **F7** keys can be used to move to the start and end of a page respectively.

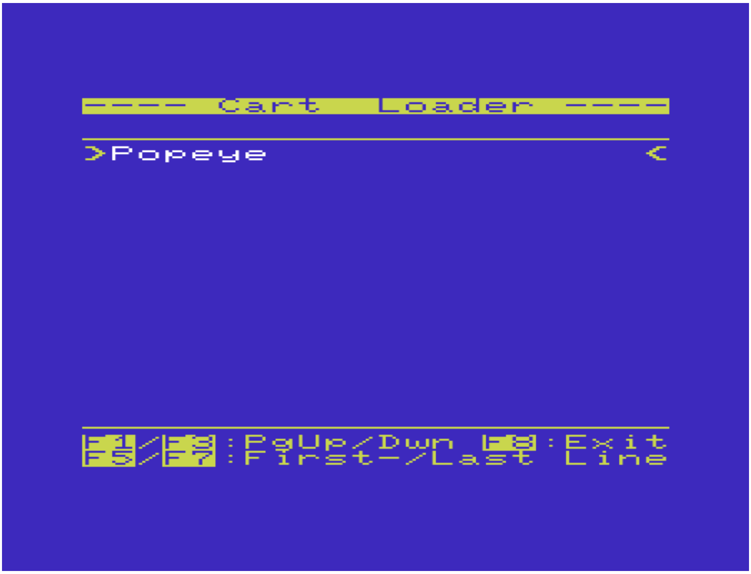
Cartridge Loader

This submenu allows memory configurations to be set up and program files to be loaded from flash by selecting an entry from a list. In order to add a selection the following steps must be taken

1. Define the memory configuration, file loading and launching commands in a loader file (see “Loader Files”)

2. Copy the definition to flash (see “Saving to Flash”)

Once the definition has been written to flash the program can be loaded and launched without needing the SD card to be present.



```
----  Cart  Loader  -----

>Popeye                 <

F1/F3:PgUp/Dwn F8:Exit
F5/F7:First-/Last Line
```

The menu is navigated by using either the cursor keys or the joystick, an entry is selected by pressing **RETURN** or the fire button. When there is more than one page of entries the **F1** and **F3** keys can be used to move between pages. The **F5** and **F7** keys can be used to move to the start and end of a page respectively.

Secure Digital Drive

The SD2IEC drive allows Secure Digital (SD) cards to store files which can then be accessed by the VIC-20 in the same way as would if they were on a floppy disk.

As with other devices attached to the serial bus the SD2IEC responds to commands sent to it on the command channel. This can be done using the @ wedge command or the **OPEN** and **PRINT#** BASIC commands. The following sections describe the various commands that are implemented.

DOS Commands

The following commands perform operations in the same manner as physical disk devices. Only the first character of a command needs to be given, a **:** (colon) must be present between the command and any parameters. Further details of these commands can be found in books such as “1541 User’s Guide”.

SCRATCH

Deletes one or more files given as parameters. Multiple file names must be separated by commas, wildcard patterns containing ***** and **?** may also be used.

RENAME

Changes the name of a file, the existing and replacement file names are specified as **NEWNAME=OLDNAME**.

COPY

Creates a duplicate of a file, the new and existing file names are specified as **NEWNAME=EXISTINGNAME**.

If multiple existing files are specified then the new file will contain the concatenation of them, for example **NEWFILE=START,MIDDLE,END**.

NEW

Erases the contents of a disk image (see below). The disk name and ID must specified as **NAME,ID**.

This command cannot be used to erase an entire SD card.

Container Files

In order to record the properties of a file that is written directly to the SD card a container file is used. The suffix of a container file indicates the type

Suffix	File Type
.Pxx	Program (PRG)
.Rxx	Relative data (REL)
.Sxx	Sequential data (SEQ)
.Uxx	User defined (USR)

The final two characters are numeric, usually **00** unless the file name on the SD card is identical with another file.

Each container file has a 26 byte header before the actual file contents.

Disk Images

As well as accessing files written directly to the SD card the SD2IEC can handle image files that represent an entire floppy disk. These images have track and sector geometry identical to various physical disk devices

Suffix	Drives	Capacity
.D64	4040, 1541 etc	170K
.D71	1571	340K
.D81	1581	800K

To activate a disk image the **CD** command is used, for example **CD:GAMES.D64**. Once active all further disk operations affect the disk image rather than the SD card. To leave a disk image and return to the SD card the command **CD:**← is used.

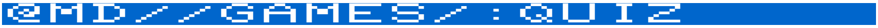
Subdirectories

Subdirectories (directories within directories) may be created, navigated into and deleted on the SD card. Directories may be referred to as follows

- Relative to the current directory, for example **GAMES**
- A directory relative to the current directory, for example **/GAMES/ARCADE**
- Relative to the root of the SD card, for example **//GAMES/ARCADE**

Creating Directories

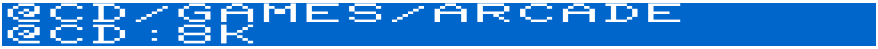
The **MD** command creates an empty directory according to the form of the parameter given. Except when creating a directly below the current directory the new directory name must be preceded with a **:**



```
@MD//GAMES/:QUIZ
```

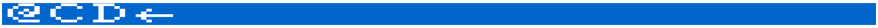
Changing Directory

The **CD** command make the directory specified by parameter the current directory for future disk operations



```
@CD/GAMES/ARCADE
@CD:8K
```

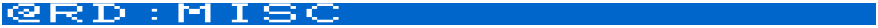
To change to the directory above the current one use the ← character



```
@CD←
```

Deleting Directories

The **RD** command removes an empty directory. Only subdirectories of the current directory can be deleted



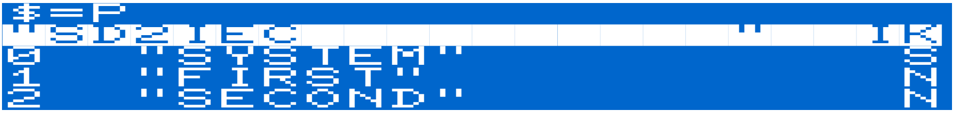
```
@RD:MISC
```

Partitions

SD cards may be contain multiple formatted partitions. Each partition is a separate logical disk and accessed in a similar manner to physical devices that have multiple drives or partitions

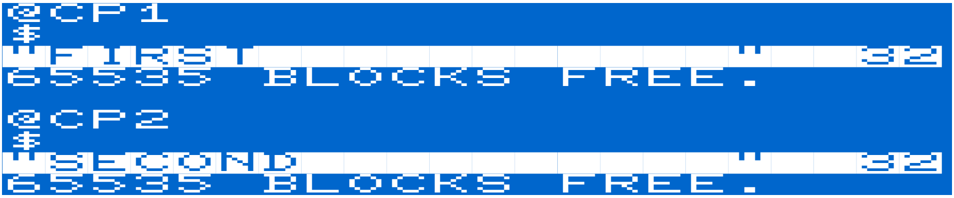
- Dual disk drives such as the 4040
- Hard disk drives such as the CMD HD-40

The list of partitions present on an SD card can be displayed using the **$=P** command



```
$=P
"SD2IEC          "  IK
0  "SYSTEM"          S
1  "FIRST"           N
2  "SECOND"          N
```

A partition can be made to be the current one using the **CP** command



```
@CP1
$
"FIRST           "  32
65535 BLOCKS FREE.

@CP2
$
"SECOND          "  32
65535 BLOCKS FREE.
```

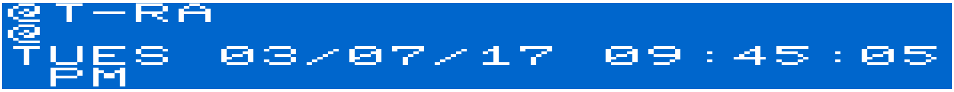
Real Time Clock

The SD2IEC device contains a Real Time Clock (RTC) which holds the current date and time. The RTC is battery-backed so that it keeps correct time even when the VIC-20 is turned off.

The following commands can be used to read the date and time and to set it.

Reading the RTC

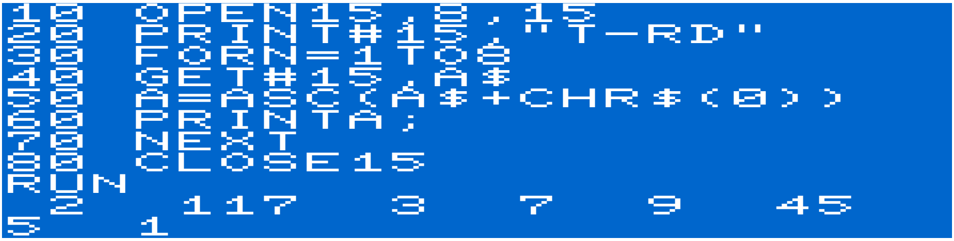
The **T-R** command reads the current time and date into the command channel. Multiple output formats are supported, the simplest ASCII, is selected by appending **A** to the command. The date is displayed in US order: *month/day/year*



```
@T-RA
@
TUES 03/07/17 09:45:05
 PM
```

Advanced Formats

The date and time can be read by program so that each component of the date and time is available as a separate value. Two formats are available: decimal and binary coded decimal (BCD), appending **D** chooses decimal and **B** chooses BCD



```
10 OPEN15,8,15
20 PRINT#15,"T-RD"
30 FORN=1TO8
40 GET#15,A$
50 A=ASC(A$+CHR$(0))
60 PRINTA;
70 NEXT
80 CLOSE15
RUN
 2  117  3  7  9  45
5  1
```

The eight fields are

1. day of the week (0 = Sunday)

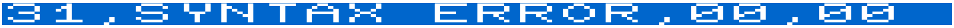
2. year (offset from 1900)
3. month
4. day
5. hour (1-12)
6. minute
7. seconds
8. AM/PM (0 = AM, 1 = PM)

The same fields are returned for the BCD format except multiple digit values are in the form of pairs of 4 bits. For example

45 would be returned as <u>4</u> * 16 + <u>5</u> = 69

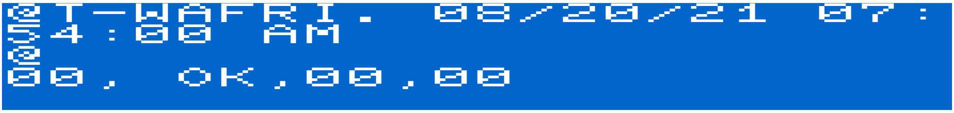
Setting the RTC

If the RTC is not initialized attempting to read it results in an error



```
31,SYNTAX ERROR,00,00
```

The **T-W** command must be used to set an initial date and time. Any of the formats described in “Reading the RTC” may be used



```
@T-WAFRI. 08/20/21 07:
54:00 AM
@
00, OK,00,00
```

The day of the week must be one of the following four character values

- SUN.
- MON.
- TUES
- WED.
- THUR
- FRI.
- SAT.

Other Commands

Many other commands are supported by the SD2IEC, some provide options to the output of other commands. For full details please refer to the README file linked from https://sd2iec.de/.

Wedge Commands

The following commands are available in immediate mode, they cannot be included in BASIC programs. They may also be used in loader programs see “Loader Files” on page 29.

The wedge interpreter usually resides in BLK5, if the “All RAM” option was selected in the “RAM Manager” menu then it will reside in the 3K region within BLK0.

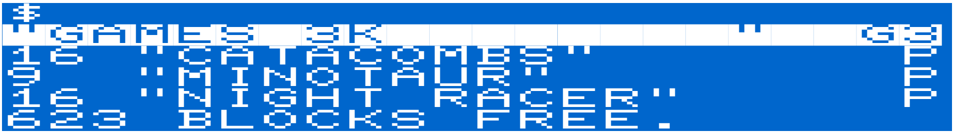
Storage Device Commands

The following commands access files stored on the SD card inserted into the SD2IEC device.

File names may be given in quotes or follow the command immediately (unless otherwise stated).

Directory Listing

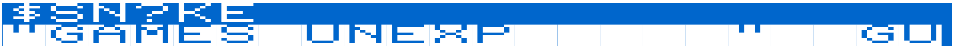
The **$** command can be used to display the contents of an SD card (or disk image or subdirectory) without overwriting any program in memory



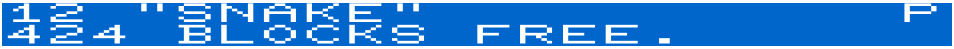
```
$
"GAMES 3K        "  G3
16 "CATACOMBS"       P
9  "MINOTAUR"        P
16 "NIGHT RACER"     P
623 BLOCKS FREE.
```

For long directory listings pressing the **CTRL** key slows down the rate at which the screen scrolls.

The list of files displayed can be reduced to those that only match a wildcard pattern. The character **?** matches any character in any position in the name

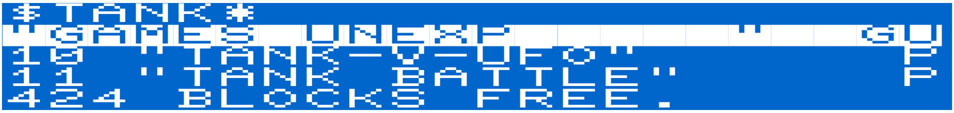


```
$SN?KE
"GAMES UNEXP     "  GU
```



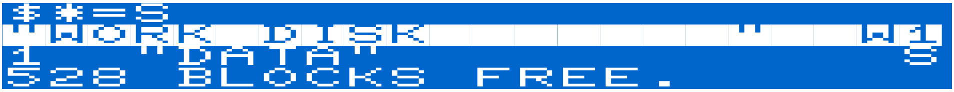
```
12 "SNAKE"           P
424 BLOCKS FREE.
```

The character ***** matches any number of characters at the end of the name



```
$TANK*
"GAMES UNEXP     "  GU
10 "TANK-V-UFO"      P
11 "TANK BATTLE"     P
424 BLOCKS FREE.
```

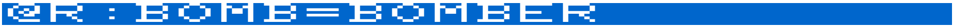
The list can also be reduced to specific file types



```
$*=S
"WORK DISK       "  W1
1  "DATA"            S
528 BLOCKS FREE.
```

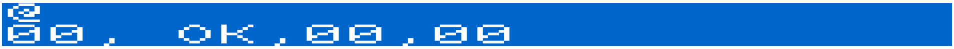
Disk Commands

The **@** command can be used to send any of the commands mentioned in “Secure Digital Drive” on page 16 to the SD2IEC



```
@R:BOMB=BOMBER
```

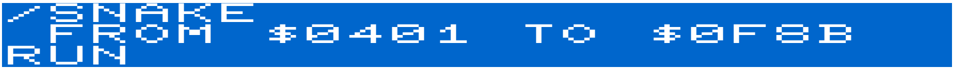
The status of the SD2IEC can be read by using the **@** command alone



```
@
00, OK,00,00
```

Load BASIC Program

The **/** command can be used to load a program from the current device

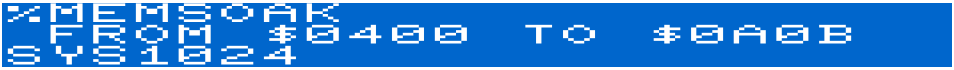


```
/SNAKE
 FROM $0401 TO $0F8B
RUN
```

The start and end addresses are displayed in hex.

Load Machine Code Program

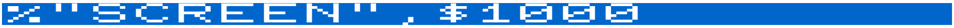
The **%** command can be used to load a machine code program into the correct location in memory



```
%MEMSOAK
 FROM $0400 TO $0A0B
SYS1024
```

The start and end addresses are displayed in hex, the start address is often used to execute the program.

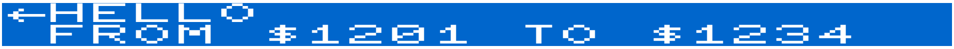
To load a program at a specific address the file name must be given in quotes followed by the address



```
%"SCREEN",$1000
```

Save BASIC or Machine Code Program

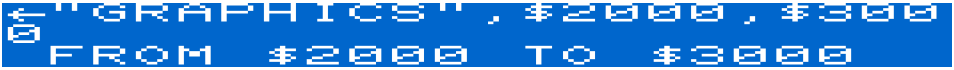
The ← command can be used to save the program in memory to the current device



```
←HELLO
 FROM $1201 TO $1234
```

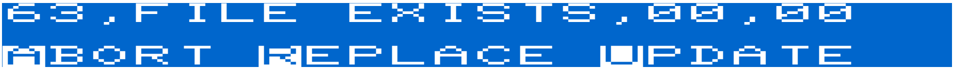
The start and end addresses are displayed in hex.

To save a machine code program the file name must be given in quotes followed by the start and end addresses



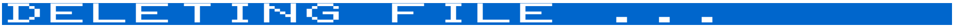
```
←"GRAPHICS",$2000,$300
0
 FROM $2000 TO $3000
```

If the file already exists on the device the following prompt is displayed



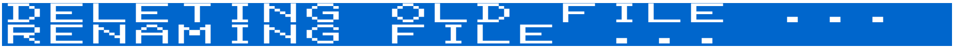
```
63,FILE EXISTS,00,00
ABORT REPLACE UPDATE
```

Pressing **R** deletes the existing file and saves the current program with the file name given



```
DELETING FILE ...
```

Pressing **U** renames the existing file with a ` (single quote) character at the beginning then saves the current program with the file name given



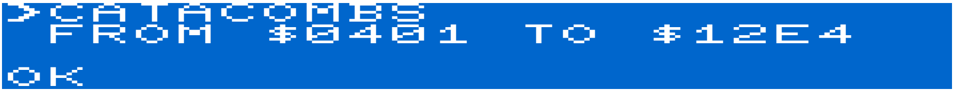
```
DELETING OLD FILE ...
RENAMING FILE ...
```

Pressing **A** cancels the operation.

This command is not available when the wedge is resident in BLK0.

Verify BASIC Program

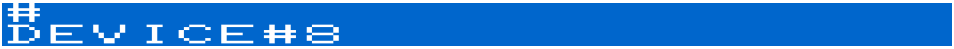
The **>** command can be used to verify that the program in memory matches the copy saved to the current device



```
>CATACOMBS
 FROM $0401  TO  $12E4
OK
```

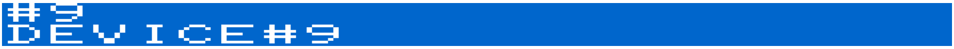
Show & Change Drive

The **#** command can be used to display the current serial device number



```
#
DEVICE#8
```

To change to another device a number between 8 and 15 may be appended



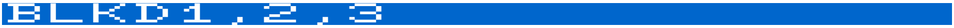
```
#9
DEVICE#9
```

Memory Block Commands

The following commands control the availability of and access to specific memory blocks.

Block Disable

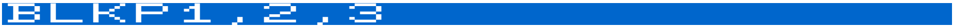
The **BLKD** command causes one or more memory blocks to be disabled. The blocks affected are appended to the command, multiple blocks are separated by commas



```
BLKD1,2,3
```

Block Protect

The **BLKP** command causes one or more memory blocks to become read-only. This can be useful if a program has copy protection to prevent it being run from RAM. The blocks affected are appended to the command, multiple blocks are separated by commas



```
BLKP1,2,3
```

I/O Register Disable

The **NOIO** command disables the I/O registers used to communicate with the FE3. This can be useful if another cartridge is present that also uses the same I/O region occupied by the FE3.

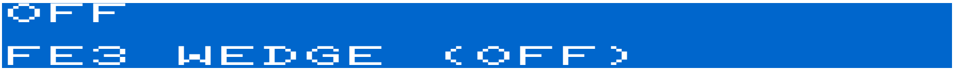
Miscellaneous Commands

System Reset

The **RESET** command can be used to restart the VIC-20. If an autostart signature is found at the beginning of BLK5 ($A000) then the cold start routine is called. Otherwise the system enters BASIC.

Wedge Disable

The **OFF** (or **KILL**) command can be used to remove the wedge from memory. Only the standard BASIC commands will be available after running this command



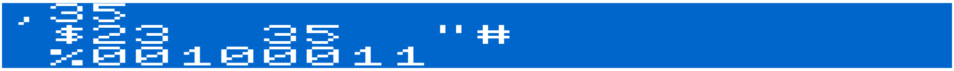
```
OFF
FE3 WEDGE (OFF)
```

Program Recovery

The **OLD** (or **UNNEW**) command can be used to restore a BASIC program that was in memory before the **NEW** command was run or the system was reset.

Number Format

The **,** command can be used to convert numbers between different number bases



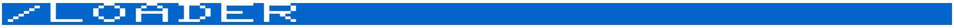
```
,35
 $23  35  "#
 %00100011
```

The **,** may be followed by a decimal, hexadecimal (prefixed with ‘$’) or a binary number (prefixed with ‘%’). An integer or floating point variable may also be given.

Loader Files

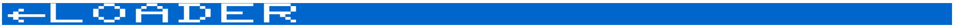
In order to define the contents of the “Disk Loader” menu a loader file needs exist in root of the SD card and in every subdirectory and image file with programs to be included.

Loader files are BASIC programs that have a specific format and structure. Loader files are always named **LOADER** and can be loaded using the **/** wedge command



```
/LOADER
```

edited and then saved using the ← command



```
←LOADER
```

Program Name

The text of the menu entry is defined by a line containing the name surrounded by **"** (double quotes). A program name may be up to 20 characters in length and may contain upper and lower case letters.

Loader Commands

The actions to be performed when the menu entry is selected all start with **+** (plus). Commands fall into four categories

1. Disk device commands
2. Commands that load the program
3. Memory configuration commands
4. Commands that launch the program

Commands are performed in the order they appear in the loader file except commands that launch the program (which are performed last).

Disk Commands

Disk commands, for example to activate a disk image, may be executed to prepare the system in some way. Commands start with **@"** and end with **"** (double quotes). Any of the commands mentioned in “Secure Digital Drive” on page 16 may be used.

Program Loading

One or more files may be loaded by giving the file name surrounded by **"** (double quotes). The program type may be given by appending a **,** (comma) and one of

- **B** – BASIC program, relocated to start of BASIC memory
- **P** – machine code program, loaded at the start address from the file (or, optionally, at an address that follows)
- **C** – cartridge, loaded at an address that follows

A load address is a hexadecimal number prefixed with **$** and is separated from the program type by a **,** (comma).

Memory Configuration

Any of the following commands described in “Memory Block Commands” on page 26 may be used to disable or protect specific memory blocks.

Program Launch

To start the program one of the following commands must be used

RUN

Start executing a BASIC program.

SYS

Start executing a machine code program, the start address is a hexadecimal number prefixed with **$**.

RESET

The autostart code of a cartridge can be triggered by performing a system reset.

To enter the menu defined in a loader file within a subdirectory or disk image the **RELOAD** command must be used to read the contents of the new file.

Comments

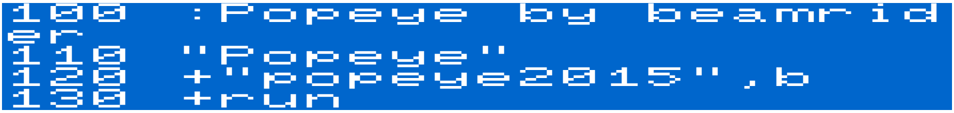
Comments may appear anywhere in a loader file. Comments start with **:** (colon) and are free-form.

Examples

The following loader file fragments illustrate some of the possible ways of loading programs. By combining the commands described above almost any action can be defined.

8K+ Expansion BASIC Program

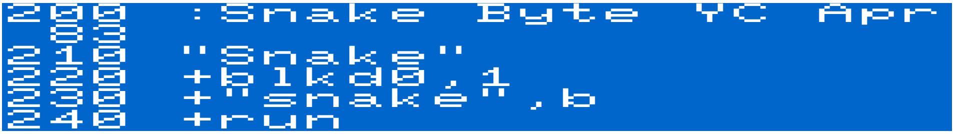
BASIC programs written for 8K (or more) expanded systems expect BASIC to start at $1200 and the screen to start at $1000. No memory configuration changes are needed, the program can be just loaded and launched



```
100 :Popeye by beamrid
er
110 "Popeye"
120 +"popeye2015",b
130 +run
```

Unexpanded BASIC Program

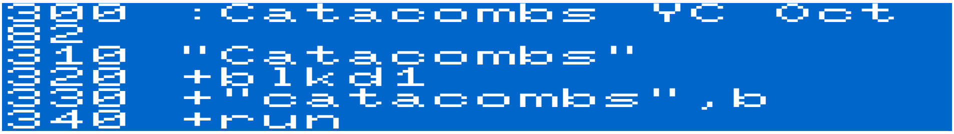
BASIC programs written for unexpanded systems expect BASIC to start at $1000 and the screen to start at $1E00. In order for the KERNAL to configure this the memory expansion in RAM1,2,3 and BLK1 need to be disabled as a minimum



```
200 :Snake Byte YC Apr
 83
210 "Snake"
220 +blkd0,1
230 +"snake",b
240 +run
```

3K Expansion BASIC Program

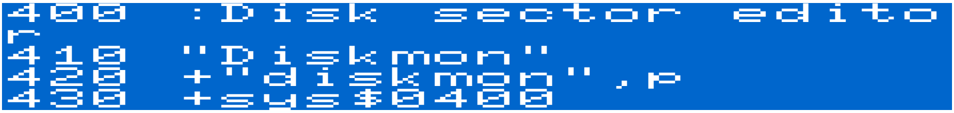
BASIC programs written for 3K expanded systems expect BASIC to start at $0400 and the screen to start at $1E00. In order for the KERNAL to configure this the memory expansion in BLK1 needs to be disabled as a minimum



```
300 :Catacombs YC Oct
82
310 "Catacombs"
320 +blkd1
330 +"catacombs",b
340 +run
```

Machine Code Program

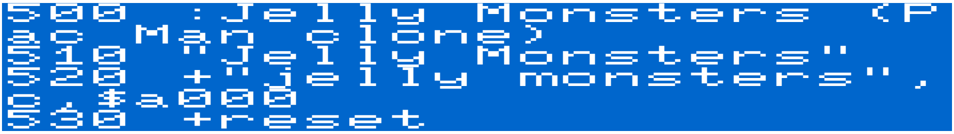
Machine code programs must be loaded into the correct memory location to work. To launch them the correct address must be passed to **SYS**, usually this is the same as the load address



```
400 :Disk sector edito
r
410 "Diskmon"
420 +"diskmon",p
430 +sys$0400
```

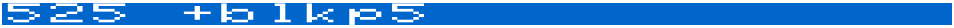
Game Cartridge

Most game and utility cartridges contain a ROM that appears at $A000 (BLK5). They automatically start when the system is powered on so are launched using a loader file with **RESET**



```
500 :Jelly Monsters (P
ac Man clone)
510 "Jelly Monsters"
520 +"jelly monsters",
c,$a000
530 +reset
```

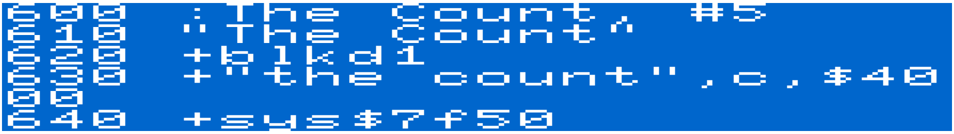
Some cartridges have a copy protection check which tries to write to the memory containing the program. To protect the memory block the **BLKP** command can be used



```
525 +blkp5
```

Adventure Cartridge

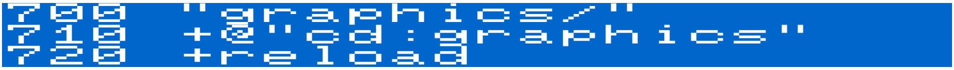
The Scott Adams Adventure cartridges are unusual: they are 16K in size, loading at $4000; they expect the screen to start at $1E00; and they have a start address of 32592 ($7F50)



```
600 :The Count, #5
610 "The Count"
620 +blkd1
630 +"the count",c,$40
00
640 +sys$7f50
```

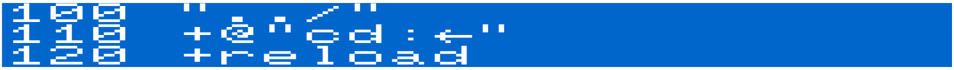
Directory & Image Navigation

In order to move into and out of subdirectories and disk images menu entries must be defined. For example if an SD card has a directory called **GRAPHICS** then in the loader file that contains it there should be the following



```
700 "graphics/"
710 +@"cd:graphics"
720 +reload
```

In the **GRAPHICS** directory the loader file should contain the following



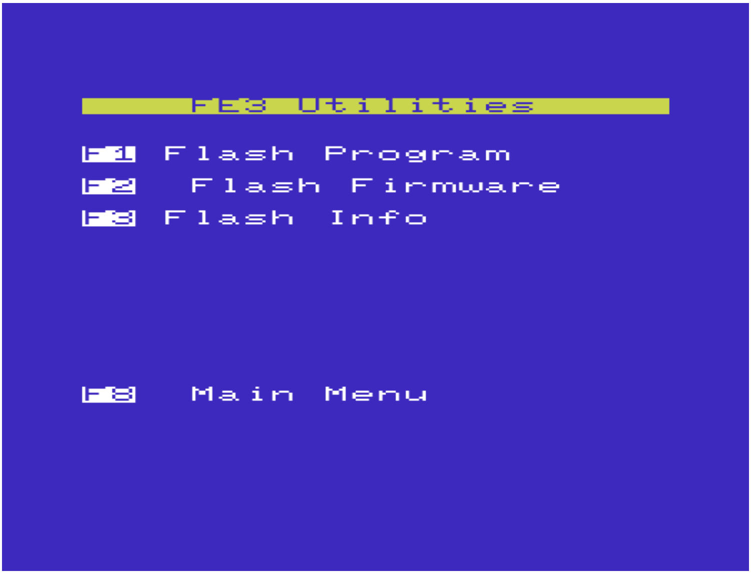
```
100 "../"
110 +@"cd:←"
120 +reload
```

Similar entries should be defined for disk images.

Saving to Flash Memory

The EEPROM can be used to store commonly-used programs so that they may be launched almost immediately. Before a program can be saved to flash a definition must be written in a loader file (see “Loader Files” above).

Flashing a program is done using the “Flash Program” option of the “FE3 Utilities” menu

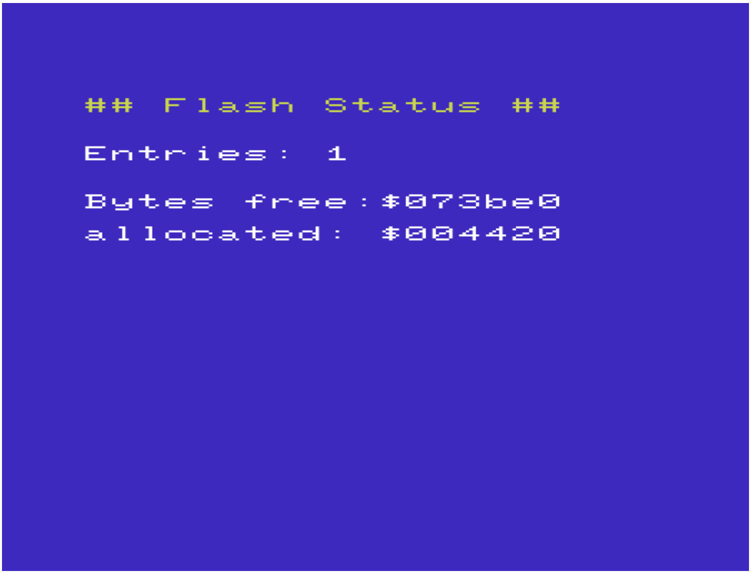


The contents of the loader file on the SD card is then displayed.

The menu is navigated by using either the cursor keys or the joystick, an entry is selected by pressing **RETURN** or the fire button. When there is more than one page of entries the **F1** and **F3** keys can be used to move between pages. The **F5** and **F7** keys can be used to move to the start and end of a page respectively.

Selecting an entry starts the process of writing the program to flash

On completion the contents of the loader file is again displayed. The amount of available flash memory is shown on the “Flash Status” screen accessed using the “Flash Info” option



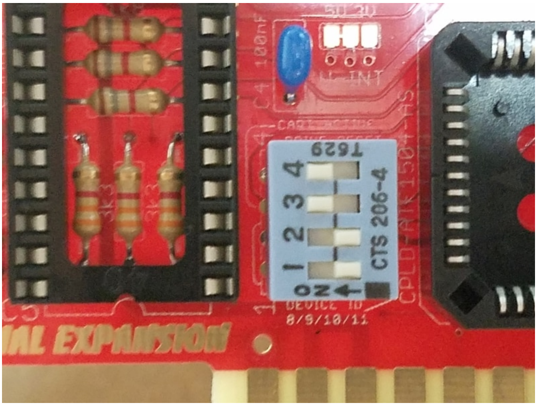
```
## Flash Status ##

Entries: 1

Bytes free:$073be0
allocated: $004420
```

Appendix A – DIP Switches

The cartridge has a set of four Dual In-line Package (DIP) switches which can be used to change how it functions in a number of areas. Depending on the case supplied these switches may not be easily accessible but they should not need changing during normal operation.



The switch functions are summarized in the table below:

Switch	Use	Default
1	SD2IEC device number: 1 off, 2 off = 8 1 on, 2 off = 9 1 off, 2 on = 10 1 on, 2 on = 11	Off
2		Off
3	Cartridge enable, SD2IEC is unaffected	On
4	SD2IEC resets when FE3 is reset	On

Warning! **The VIC-20 must be powered off before changing the switch settings.**

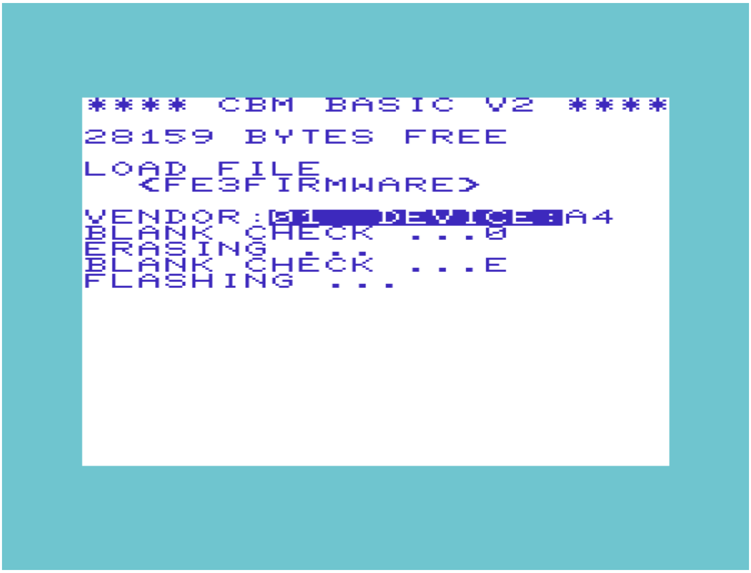
Appendix B – Updating Firmware

The firmware responsible for the menus and wedge can be updated from the “Flash Firmware” option on the “FE3 Utilities” menu.

The firmware binary should be written to the root directory of an SD card with the file name of **FE3FIRMWARE**.

Warning! The VIC-20 must not be powered off or reset while the firmware is being updated.

Warning! Updating the firmware will erase all cartridge images that have been written to flash.



```
**** CBM BASIC V2 ****

28159 BYTES FREE

LOAD FILE
  <FE3FIRMWARE>

VENDOR:01    DEVICE:A4
BLANK CHECK ...0
ERASING ...
BLANK CHECK ...E
FLASHING ...
```

Appendix C – Programmers Reference

Developers can access the memory present on the FE3 directly by programming the I/O registers described below.

Memory Map

The RAM and EEPROM devices occupy separate 19-bit address spaces.

18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bank				Base		Offset												

Base

A14	A13	Region
0	0	RAM1,2,3
0	0	BLK1
0	1	BLK2
1	0	BLK3
1	1	BLK5

Register Descriptions

Two eight bit registers control the mapping of the RAM and EEPROM devices into the VIC-20 memory address space.

All registers are initialized to zero at power on.

The register lock bit is set at power on.

Registers are accessible if both

- the lock bit is clear
- bit 7 of the resource register is clear

Mode register $9C02

7	6	5	4	3	2	1	0
Mode			Bank/Parameter				

Mode

000	Start mode
010	Super ROM mode
100	RAM 1 mode
110	RAM 2 mode
101	Super RAM mode
011	RAM/ROM
001	Flash mode

Start Mode

Region	Reads from	Read control	Writes to	Write control
RAM1,2,3	-	-	-	-
BLK1	-	-	RAM Bank 1	-
BLK2	-	-	RAM Bank 1	-
BLK3	-	-	RAM Bank 1	-
BLK5	EEPROM Bank 0	-	RAM Bank 1	-

Any write to an offset in BLK5 clears the register lock bit.

Any read from an offset in BLK5 sets the register lock bit.

Super ROM Mode

Region	Reads from	Read control	Writes to	Write control
RAM1,2,3	RAM Bank 0	-	RAM Bank 0	-
BLK1	EEPROM	Bank $b_3b_2b_1b_0$	RAM Bank 1	-
BLK2	EEPROM	Bank $b_3b_2b_1b_0$	RAM Bank 1	-
BLK3	EEPROM	Bank $b_3b_2b_1b_0$	RAM Bank 1	-
BLK5	EEPROM	Bank $b_3b_2b_1b_0$	RAM Bank 1	-

RAM 1 Mode

Region	Reads from	Read control	Writes to	Write control
RAM1,2,3	RAM Bank 0	-	RAM Bank 0	b_0 – write protect
BLK1	RAM Bank 1	-	RAM	b_1 – 0 = Bank 1, 1 = Bank 2
BLK2	RAM Bank 1	-	RAM	b_2 – 0 = Bank 1, 1 = Bank 2
BLK3	RAM Bank 1	-	RAM	b_3 – 0 = Bank 1, 1 = Bank 2
BLK5	RAM Bank 1	-	RAM	b_4 – 0 = Bank 1, 1 = Bank 2

RAM 2 Mode

Region	Reads from	Read control	Writes to	Write control
RAM1,2,3	RAM Bank 0	-	RAM Bank 0	b_0 – write protect
BLK1	RAM	b_1 – 0 = Bank 1, 1 = Bank 2	RAM Bank 1	-
BLK2	RAM	b_2 – 0 = Bank 1, 1 = Bank 2	RAM Bank 1	-
BLK3	RAM	b_3 – 0 = Bank 1, 1 = Bank 2	RAM Bank 1	-
BLK5	RAM	b_4 – 0 = Bank 1, 1 = Bank 2	RAM Bank 1	-

Super RAM Mode

Region	Reads from	Read control	Writes to	Write control
RAM1,2,3	RAM Bank 0	-	RAM Bank 0	-
BLK1	RAM	Bank $b_3b_2b_1b_0$	RAM	Bank $b_3b_2b_1b_0$
BLK2	RAM	Bank $b_3b_2b_1b_0$	RAM	Bank $b_3b_2b_1b_0$
BLK3	RAM	Bank $b_3b_2b_1b_0$	RAM	Bank $b_3b_2b_1b_0$
BLK5	RAM	Bank $b_3b_2b_1b_0$	RAM	Bank $b_3b_2b_1b_0$

RAM/ROM Mode

Region	Reads from	Read control	Writes to	Write control
RAM1,2,3	RAM Bank 0	-	RAM Bank 0	b_0 – write protect
BLK1	RAM/EEPROM	b_1 – 0 = RAM Bank 1, 1 = EEPROM Bank 0	RAM	b_1 – 0 = Bank 1, 1 = Bank 2
BLK2	RAM/EEPROM	b_2 – 0 = RAM Bank 1, 1 = EEPROM Bank 0	RAM	b_2 – 0 = Bank 1, 1 = Bank 2
BLK3	RAM/EEPROM	b_3 – 0 = RAM Bank 1, 1 = EEPROM Bank 0	RAM	b_3 – 0 = Bank 1, 1 = Bank 2
BLK5	RAM/EEPROM	b_4 – 0 = RAM Bank 1, 1 = EEPROM Bank 0	RAM	b_4 – 0 = Bank 1, 1 = Bank 2

Flash Mode

Region	Reads from	Read control	Writes to	Write control
RAM1,2,3	-	-	-	-
BLK1	EEPROM	Bank $b_3b_2b_1b_0$	EEPROM	Bank $b_3b_2b_1b_0$
BLK2	EEPROM	Bank $b_3b_2b_1b_0$	EEPROM	Bank $b_3b_2b_1b_0$
BLK3	EEPROM	Bank $b_3b_2b_1b_0$	EEPROM	Bank $b_3b_2b_1b_0$
BLK5	EEPROM	Bank $b_3b_2b_1b_0$	EEPROM	Bank $b_3b_2b_1b_0$

Resource register $9C03

7	6	5	4	3	2	1	0
e	a	b	5	3	2	1	r

e – Register disable

0	Registers enabled
1	Registers disabled

ab – Invert address lines

a	Invert A14
b	Invert A13

5 – BLK5 disable

0	BLK5 enabled
1	BLK5 disabled

3 – BLK3 disable

0	BLK3 enabled
1	BLK3 disabled

2 – BLK2 disable

0	BLK2 enabled
1	BLK2 disabled

1 – BLK1 disable

0	BLK1 enabled
1	BLK1 disabled

r – RAM1,2,3 disable

0	RAM1,2,3 enabled
1	RAM1,2,3 disabled