A
VERGECOURT
INTERFACE
PRODUCT

COPYRIGHT NOTICE

If you have any questions about this copyright, please contact:-

Vergecourt Ltd.,
17, Nobel Square,
BASILDON,
Essex.
SS13 1LP.

.....

Any Software House incorporating SUPER HIDOS in programs for sale to third parties must enter into a license agreement with Vergecourt Ltd.

TABLE OF CONTENTS

OPERATING INSTRUCTIONS :

APPENDICES :

INTRODUCTION

We would strongly advise that you read this section and also chapters 1 & 2 before installing your RAMEX card.

The RAMEX 128 is a plug-in printed circuit board for the Apple II and Apple II plus, that contains 128K bytes of additional random access memory (Ram).

The manufacturing process employed by Vergecourt Ltd., and pioneered with the RAMEX 16, has ensured that your RAMEX 128 contains the very latest in technology to produce the finest closed track engraved, epoxy sealed board possible and, of course, utilises 64K Ram chips to maximise its performance.

The RAMEX 128 is a multipurpose expansion card and is completely machine compatible with all existing expansion boards, and can be used to expand the user memory of standard software packages like Visicalc, to a maximum of 136K with the SUPER EXPANDER utility program.

The RAMEX 128 is supplied complete with a comprehensive memory management system which includes Disk Emulation Software and DOS Relocation. This together with a powerful set of additional DOS commands will allow you to write or convert your existing software to make use of the dramatic decreases in access times now possible.

The RAMEX 128 continues the policy of Vergecourt to utilise all your existing hardware and the supplied software makes full use of any existing 16K expansion board(s) you may have fitted into your system.

CHAPTER 1

Installation & Handling

Before installing the RAMEX 128 Memory Expansion Board, please read the chapter "which slot", this gives guidance in deciding in which slot to place your RAMEX.

To install the RAMEX 128, you will simply plug the card into a socket inside the computer. Please read these instructions thoroughly before carrying out the installation:-

1/ Turn off the power switch at the back of the computer and also make sure any Disk Drive unit connected is turned off. This is important to prevent damage to the computer. If the power is left on, removal or insertion of any card could cause permanent damage, to both the RAMEX 128 and any other device connected to the computer.

2/ Remove the top cover from the computer. This is done by pulling up on the cover at the rear edge, (the edge farthest from the keyboard), until the two corner fasteners come apart. Do not continue to lift the rear edge, but slide the cover backwards until it comes free.

3/ Inside the computer, across the rear of the circuit board, there is a row of eight long, narrow sockets, called "slots". The leftmost one (looking at the computer from the keyboard end), is slot Ø and the rightmost is slot 7, which is normally covered by a sticky paper cover, to prevent a card being inserted. This slot, No. 7 is normally reserved for a colour card This leaves slot Ø - 6 and the RAMEX 128 can be installed in whichever of these slots you have chosen.

4/ Most printed circuit boards contain IC's (Integrated Circuits) commonly called chips. It is possible to damage these chips by discharging "static" electricity through them. You can quite easily do this if you have static electricity on your hands. Thus it is essential to "Earth" your hands, to remove any static electricity before handling any P.C.B. including the RAMEX 128. This is easily accomplished by touching the metal casing of the power supply located to the left of the inside of the computer.

5/ Be sure THE POWER IS OFF TO THE COMPUTER AND ANY CONNECTED DISK DRIVE UNIT, BEFORE YOU INSERT OR REMOVE ANY CARD FROM THE COMPUTER. Insert the "fingers" portion of the RAMEX 128 memory expansion card into the free slot you have chosen. The "fingers" portion will enter the socket with some friction and will then seat firmly, but make sure it is seated evenly in the slot and not with one end up or down. Since the fingers make the electrical contact, it is a good idea to keep your fingers from touching them. Before installation, you may wish to use rubbing alcohol to clean the fingers on the board.

6/ Other Ram extension cards made by peripheral manufacturers require you to remove the "48th K chip" on the motherboard, and insert a chip connected to the Ram card by a ribbon cable and header. This is not necessary with the RAMEX 128 memory expansion board and is one of its great advantages and means that it also can be moved from slot to slot or another computer very easily.

7/ Replace the cover of the computer, remember to start by sliding the front edge of the cover into place press down on the two rear corners until they pop into place.

8/ The RAMEX 128 memory expansion board is now installed and you may now turn on your computer.

9/ Keep the packaging in which your RAMEX 128 was supplied in a safe place. In the unlikely event of your having to return your card for service, its purpose designed packaging will keep it safe in transit.

10/ Consult chapter 3 for the instructions of how to use your supplied utility software.

CHAPTER 2

"Which Slot?"

The answer depends on the use to which you want to put your RAMEX 128 board(s) and to help you to decide which is the most appropriate slot, we have given examples of some typical system configurations.

Example 1.

RAMEX 128 for Visicalc expansion, also requiring 16K expansion for Pascal Ledger System and 16K expansion to run 56K CPM Wordstar Word Processing system.

Typical System Configuration

Slot Ø RAMEX 128
1 Printer Card
2 Spare
3 80 Column Card
4 Z80 Softcard
5 Spare
6 Disk Drive Controller
7 Spare

The main use to which the RAMEX 128 will be put in the System Configuration above, is to run Visicalc with 136K of user memory. This is accomplished in this instance with the RAMEX 128 in slot Ø and uses the SUPER EXPANDER utility Disk, which is part of the Vergecourt range and would need to be purchased separately.

As the RAMEX 128 has been designed to be as flexible as possible, it is also able to default to a standard 16K expansion card when necessary, and when installed in slot Ø, provides the required memory expansion to run Pascal programs and also 56K CP/M in conjunction with the Microsoft Z80 Softcard.

In this configuration when used with the utility software provided, the RAMEX 128 can be loaded with Integer in the first block, DOS on the second block, with the remaining 96K available as a virtual Disk. This configuration is ideal for program development work, where the programmer can store his basic and machine code program utilities for almost instantaneous storage and retrieval.

Example 2.

RAMEX 128 to be used as virtual memory Disk Drive, Visicalc displaying 136K of user memory, Integer and relocated DOS using existing 16K Ram extension cards.

Typical System Configuration

Slot Ø Apple Language Card (16K Ram Card)
1 Printer Card
2 Spare
3 RAMEX 16
4 Spare
5 RAMEX 128
6 Disk Drive Controller
7 Euro Colour Card

In this configuration, the main use of the system is for the RAMEX 128 to be used as a virtual memory Disk Drive. Using the utility software provided, the RAMEX can be used in direct mode by copying programs to it using 'COPYA' or *INIT and then 'FID' and then 'RUN'ing, 'LOAD'ing or 'SAVE'ing etc. The above is sufficient for program development work, but when an applications program is envisaged, a far more sophisticated approach is required and we have incorporated 12 additional DOS commands, so that all Disk access etc., is carried out either under program control or in indirect mode, (please see chapter 6 for a detailed explanation of this).

In this example configuration, we are assuming that you have already two 16K expansion cards, for example, an Apple language card in slot Ø and a RAMEX 16 in say slot 3. The utility software enables you to load Integer (Applesoft), to the expansion card in slot Ø and relocate DOS to the RAMEX card in slot 3, leaving approximately 44K of Motherboard Ram for program space whilst retaining the RAMEX 128(s) as virtual memory Disk Drive(s). For the more technically minded, the enhancements to DOS are contained with the 'Interface to DOS' above relocated Hi-mem as shown in appendix 'B', but this is explained in more detail in chapter 3.

Summary

As you can see from the examples given, the RAMEX 128 is a very versatile memory expansion card. As Vergecourt have retained the feature of carrying out the 'Ram Refresh' function on-board, the RAMEX 128 can easily be installed in any slot, its use and function largely being determined by the user.

The RAMEX 128 will function as a Disk Drive in slot 1 - 6, using the supplied utility software and when placed in slot Ø, as a standard 16K expansion card (for pascal - CPM etc). Alternatively, it can have Integer and DOS loaded, leaving 96K of virtual memory or Disk Drive.

Using the SUPER EXPANDER utility, the RAMEX 128 can be used in any slot to give 136K of user memory with Visicalc, (16 sector only). This program is intelligent enough to find where the RAMEX 128 cards are located in your system without any form of user prompt.

CHAPTER 3

Using the RAMEX 128

Dependant upon the configuration of your system, the RAMEX 128 may change some procedures you currently use to run your existing software.

The differences occur if you have installed either a 13 sector Disk system with the Microsoft Z80 softcard, or you are using an Apple firmware card in slot Ø. Please skip section 1 if the above paragraph does not apply to you.

SECTION 1.

DOS 3.2 13 sector & firmware card system limitations.

The Apple firmware card (Integer or Applesoft) enables the Apple II and Apple II plus (Europlus), computers to use the alternate Basic, which was not supplied in Motherboard Rom with your computer.

If you install the RAMEX 128 in slot Ø, you will have to remove the firmware card, which means you will lose the alternate Basic.

As the RAMEX 128 was designed with no strap and header to the "48th K chip", it is quite easy to remove it and replace your firmware card when required, but this solution, at best is clumsy and a more permanent solution would be to update your system to DOS 3.3. This configuration would give you both Basics available as the systems master would load the alternate Basic to your RAMEX 128 at system boot. You could then change between Basics by typing INT and FP as necessary. Should you wish to retain your firmware card, this is possible providing a 16 sector controller is used,

thus if you are using a DOS 3.2 system, you would of course need to replace your 13 sector PROMS on your controller card which are supplied with the DOS 3.3 upgrade kit. If any of the above is unclear, or you need help in carrying out the DOS 3.3 upgrade, consult your local Apple dealer for further advice.

CP/M compatibility

If you are using the RAMEX 128 in slot Ø, (in its 16K default expansion mode), with the microsoft Z80 softcard, then you will need to use the CP/M 56 utility to enable you to use the first Block of the RAMEX 128. This utility is only found in the 16 sector CP/M master supplied with the softcard and you will, of course, need your controller card configured for Apple DOS 3.3 (16 sector) to run CP/M 56.

Software compatibility

The RAMEX 128 when installed in slot Ø and used in its 16K default expansion mode, is completely compatible with the following software.

Applesoft Basic	CP/M which allows you to use	
Integer Basic	Microsoft's	Cobol 80
Visicalc	"	Fortran 80
Apple Pascal System	"	Basic Compiler
Apple Fortran	"	Assembly Language
Apple Pilot	"	Development System.

SECTION 2.

The Enhancer "System Diskette"

The arrival of the RAMEX 128 now means that you can use the memory available as a virtual memory Disk Drive, using the SOLIDOS on your utility Disk provided, and the name of this Disk is the "ENHANCER".

Utilising the RAMEX 128 as a 'RAM Drive', access times can be decreased by up to 300% and we also offer the facility to re-locate the DOS onto an existing 16K expansion card in the system. If you are fortunate enough to have two 16K expansion cards, (or an Apple language card and one additional 16K card), one of which is situated in slot Ø, then we can offer Integer on slot Ø, relocated DOS on your second 16K card and SOLIDOS, on the RAMEX 128. We can as you see, cater for almost all possiblities.

The ENHANCER Disk provided with your Ramex 128 does not contain DOS. The program it contains will convert any ordinary Master Diskette into a SUPER HIDOS Master containing SOLIDOS.

Because the System Master Diskette that was supplied with your Apple Disk Drive contains many useful utility programs we suggest that you make a copy of it and use the resultant Diskette as the first one to be turned into a SUPER HIDOS Master, containing SOLIDOS. As explained in section (4) later, the ENHANCER will amend these utility programs where necessary, so that they can take full advantage of the increased memory available, if you are using the RAMEX 128 in slot Ø, or if you have any 16K Ram cards configured into the system.

Thus the correct sequence to use the ENHANCER, is as follows:-

(i) Boot DOS (DOS 3.3). (The ENHANCER does not contain DOS).

(ii) Insert the ENHANCER in the Disk Drive and 'boot' by typing PR#6. If it is in another slot, then you must use the appropriate slot No.

(iii) The ENHANCER will then ask you which type of HIDOS Master you require. The various types possible, are described in section (3) but for the present, we suggest you use the default option, 'Type 1', which you obtain by just pressing RETURN.

(iv) You will now be asked to insert a Master Diskette in the Disk Drive. Please note that this must be the same Drive as the one in which you booted the ENHANCER. We suggest the first Disk used is a copy of the Apple Systems Master.

You now have the opportunity to 'exit' by pressing ESC or continuing, by pressing RETURN.

(v) The ENHANCER will now convert the DOS on the Master Diskette inserted to a SUPER HIDOS Master with SOLIDOS. This is a very quick process and you can then proceed to convert any utility programs at your discretion. This is quite a lengthy operation, as there is a lot of Loading and Saving to be done, so do not be alarmed as the Disk Drive whirrs away.

(vi) Finally, you are given the choice of using the ENHANCER to make more SUPER HIDOS Masters or booting the SUPER HIDOS Systems Master you have just made, or returning to Basic.

The RAMEX 128 utility program

A short utility program is contained on the ENHANCER. After enhancing a copy of your System Master, you may wish to "transfer" this program to your new Disk and of course run it.

This program was only included to give a "brief demonstration" of some of the more Basic commands now available with "SUPER HIDOS". You may find it useful to attach it to the end of the HELLO programs of your own work Disks to quickly load up your RAMEX 128 at the beginning of a work session.

SECTION 3.

Types of Master available

IMPORTANT NOTE

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * *
* Irrespective of the Type of SUPER HIDOS Master made *
* as described below, if the RAMEX 128 is placed in   *
* slot Ø, then 'Type 1', is ignored and DOS is         *
* automatically loaded to Block 1, leaving Block       *
* Ø available for your alternate Language (ie,         *
* Integer), and the remaining 96K as a virtual         *
* memory Disk Drive. With each Type selected, the      *
* DOS enhancements are loaded to enable the RAMEX      *
* 128 to be used as a virtual memory Disk Drive.       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * *
```


TYPE 1. Default option

Will not put DOS onto a 16K expansion card in slot Ø, if there is another slot available, (so the slot Ø card is available for Integer). If the only 16K expansion card is in slot Ø, then this is where DOS will be loaded, with the DOS enhancements and DOS Interface in ordinary Ram, leaving approximately 45K of user Motherboard Ram available. If no 16K cards are found, then DOS and the DOS enhancements will be loaded in the normal way to Motherboard Ram.

TYPE 2.

Will not put DOS onto a 16K expansion card in slot Ø, (as this is always reserved for your alternate language ie, Integer). If a 16K card is found in any other slot then this is where DOS will be loaded, with the DOS Interface and DOS enhancements loaded to Motherboard Ram. If no 16K card available, other than slot Ø, then DOS and the DOS enhancements, will be loaded to Motherboard Ram, in the normal way.

TYPE 3.

Will load DOS onto the first 16K expansion card it comes to, starting at slot Ø, with the DOS Interface and DOS enhancements loaded to Motherboard Ram. If no 16K card is found, then DOS and the DOS enhancements will be loaded to Motherboard Ram in the normal way.

Note:

This very complex slot search and indentification of peripheral cards, is incorporated in the HIDOS boot process now available on any Disks you have turned into HIDOS Masters.

HIDOS compiles a table of where you have your expansion cards located in your Apple and then applies the selection criteria for re-locating DOS and the SOLIDOS enhancements. If no 128K card is found, then DOS is booted in the normal way.

NB. The DOS enhancements occupy about 1K and the DOS Interface about 2K then page Ø, stack, buffers and text page 1 occupy a further 2K leaving about 44000 bytes of 'free memory'.

The modified Master Create program on your SUPER HIDOS System Master, will create a new SUPER HIDOS Master of the same 'type' as your SUPER HIDOS Systems Master. To change the 'type' of a Master Diskette, you must use the ENHANCER Diskette as previously described.

SECTION 4.

Amendments to programs on the System Master

As noted earlier, there are many useful programs on the DOS Systems Master. Unfortunately, some of these programs will not work without modification, because they are not sufficiently intelligent enough to realise that there is now extra Ram available. This is why we suggest that the first Diskette you convert with the ENHANCER, should be a copy of your Systems Master. The ENHANCER will search through the Disk for all the utility programs that are normally on a Systems Master, and adjust them to operate successfully under SUPER HIDOS and SOLIDOS.

Examples of programs that require amendments are Master Create, FID, Renumber etc.

If for any reason you do not wish to avail yourself of this automatic facility, it does not matter. Just answer "N" to the offer made on the ENHANCER menu, for this function.

WARNING

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* Do not ask the ENHANCER to amend programs (on a Disk-     *
* ette), which have the same names, but which differ        *
* from the standard programs found on the Systems Master.   *
* They will be corrupted!!!                                 *
*                                                           *
* Additional Note                                           *
*                                                           *
* Please note that the "HELLO" and "APPLESOFT" programs     *
* on the standard System Master will not work with          *
* HIDOS. If you use the automatic modification referred     *
* to above, all will be well!                               *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

SECTION 5.

Auto Boot

There is no need to use the ENHANCER once you have made your SUPER HIDOS Master, or Slave Diskettes. Just insert your Diskette in the Disk Drive and boot. SUPER HIDOS and SOLIDOS will automatically be available, provided your expansion cards are installed. Thus for those who have the Auto-start F8 Monitor Rom, the full switch on and run facility is retained.

WARNING

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* If you 'INIT'ialise a Diskette to create a SLAVE,      *
* then this SLAVE Diskette will only boot if the         *
* configuration of the system is the same as when the    *
* Master Disk itself was made.  If the system config-    *
* uration is changed or your want to make Disks that     *
* will work with any system configuration, then you      *
* must use the "Master Create" program on your SUPER     *
* HIDOS System Master to convert your SLAVES into        *
* Masters from which new SLAVES can then be made.        *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```


CHAPTER 4

Super Expander for 136K Visicalc

Introduction

The well acclaimed Visicalc package by Personal Software Inc. has become a world beating software product, and very few Apples used in a business environment, are without this excellent program.

Vergecourt Ltd., has now combined the performance capability of the RAMEX 128 with the SUPER EXPANDER, to enable Visicalc to be displayed with 136K of user memory. You will of course need to be in possession of a DOS 3.3 (16 sector) Visicalc as supplied by Personal Software.

The SUPER EXPANDER puts three new commands into the Visicalc storage command area, which of course, is accessed by the /S command.

The commands have been added to speed up the method of accessing/saving data, because when using Visicalc with the RAMEX 128, the standard commands were found to be very slow in operation. The whole point of using the RAMEX 128, is to obtain both power and speed, and the new commands help restore the balance and you can now "LOAD/SAVE" a Visicalc model of say 110K in only TWENTY SECONDS.

New SUPER EXPANDER commands

The new commands are /SE, /SK and /SM and we will now discuss each command in detail.

(i) Loading a complete Data Disk

/SM is a new command to 'M'ount, via a Binary DUMP, a complete Data Disk into Visicalc, but only from a Disk that has previously had Data 'SAVE'ed to it via a /SK command. A model using say 112K of memory, does not of course occupy a complete Data Disk, but for speed and simplicity the /SM loads a complete Disk which therefore, should only contain one model.

Several smaller models can of course still be stored/loaded on a normal Visicalc Data Disk, with the /SS and /SL commands, which has been 'INIT'ialised with the usual /SI commands.

(ii) The New 'INIT' command

The /SE command is to 'E'xpand a Data Disk for use with the RAMEX 128, and you should use this command to 'INIT'ialise a Data Disk which is to be used in conjunction with the new commands /SM and /SK.

When 'INIT'ialising a blank or re-circulated Disk with the /SE, the command will incorporate the Disk name 'EXPANDER DISK', To check whether or not a Disk you are using is one 'INIT'ialised by the /SE command, try the following.

Issue the command /SM and press RETURN. The Drive will whirr and the Disk name will appear in the line just below the command area. If the Disk was 'INIT'ialised with the command /SE, then the name 'EXPANDER DISK' will appear and tell you it is a blank 'INIT'ialised Disk, ready for use with the /SK command.

If you have inserted an ordinary Data Disk 'INIT'-ialised with the standard /SI command, then no name will be found, which informs you it is not an 'EXPANDER DISK'. (Warning: only 'M'ount a Data Disk with Data and not a blank).

(iii) Saving a complete SUPER EXPANDER model

The /SK is a new command to keep/save Data back to a floppy Disk. This command is effectively a 'Binary Data Dump', back to a floppy Disk and will unconditionally overwrite any existing Data files. We suggest that you first check that the 'EXPANDER DISK' that you are going to use with the /SK command, does not contain an existing '128 Data file', that you wish to keep, because if you proceed with the /SK command, it will unconditionally overwrite any existing Data files. The /SK command will only 'write' to an 'EXPANDER DISK', so if you cannot 'DUMP' your Data to a Disk, then you are not using a Data Disk 'INIT'ialised by the /SE. This was incorporated as a protection device to help prevent the accidental loss of Data files. The /SK command must be used prior to a /SM (see /SM).

For the more technically minded, the SUPER EXPANDER also alters the memory space available for the internal 'Index' of Visicalc. This was found to be necessary as with the extra memory now available with the RAMEX 128, you might have been given an out of memory error whilst building a large well spaced 'Model', while the memory indicator in the command area of Visicalc, showed plenty of memory available, and the same is true as you may have found using Visicalc, with only 34K of memory, hence our alterations. Thus SUPER EXPANDER automatically restores the balance in memory reserved for the Index and Data.

Hints & Tips

When using the SUPER EXPANDER, the time Visicalc takes to recalculate between cursor movements, can seem a fraction slow. This is due to several factors.

(i) The speed of the 6502 processor which is fixed!
(ii) Re-building the Index every time the model is expanded, ie, every time the bottom right co-ordinate is increased, (Visicalc has to keep a record of where everything is on the screen).
(iii) The fantastic number of re-calculations Visicalc has to perform with every entry.

To speed up the entry of Data to a Visicalc model try the following:-

(i) Take a guess at the 'value' of the bottom right hand co-ordinate, for your proposed model, and put an entry however meaningless in this field, to establish the size of the model and the 'Index'. This will save Visicalc rebuilding Index each time you extend the bottom right co-ordinate.
(ii) 'Switch off' the automatic re-calculation function, via /GR routine, selecting 'M' for manual. This means you can build your model quicker and you only need to start the model 'working' when it is finished, or perhaps half way through, when you wish to check the mathematics of what you have done so far.

The above should speed up your Data entry with Visicalc in any memory configuration, but is especially helpful when working with models of over 100K of memory.

Important Note

The SUPER EXPANDER has the ability to work with the following 16 sector DOS 3.3 versions of Visicalc in current usage.

VC - 193BØ - AP2
VC - 202BØ - AP2
VC - 208BØ - AP2

Please consult your dealer or Vergecourt Limited if you have a different version of Visicalc.

Due to the complex protection methods incorporated by Visicalc, the Visicalc program will not boot with a Ram card (16K, 32K, 64K 128K etc), in slot 2. You can of course use slot 2 if you always boot using the SUPER EXPANDER.

CHAPTER 5

Using SOLIDOS

After using the ENHANCER Diskette as described in chapter 3, you have now created a Master Diskette that contains both HIDOS and SOLIDOS.

SOLIDOS contains all the DOS amendments necessary, to enable your RAMEX 128 to function as a virtual memory Disk Drive. Whilst all the standard DOS commands are preserved and can be used with the RAMEX 128, we have added some DOS enhancements to facilitate the use of the RAMEX 128 under program command.

To demonstrate the use of SOLIDOS:

(i) Boot your new HIDOS Master.

(ii) Type NEW (RETURN).

(iii) Type 10 REM HELLO (RETURN).

INIT HELLO, Sn (RETURN) where N = RAMEX 128 slot.

(iv) Almost instantly your cursor returns showing that the RAMEX 128 has been "formatted" and to see what is written on the Disk, you can now type "CATALOG".

Hey Presto, an instant Disk catalog, showing the normal 2 sector "HELLO" program

(v) You could now copy from your floppy Diskette to the RAMEX 128 using "COPYA" or "FID".

Important Note

Experienced programmers may well ask why we did not automatically 'INIT'ialise the RAMEX 128 at system boot, to avoid having to manually 'INIT' as explained above.

Of course we could have done this, but a closer examination, will show that this is not always desirable.

Consider the situation where you have files stored on your RAMEX 128, but you manage to corrupt DOS. The situation can easily be retrieved by replacing a Master Diskette and Typing PR#6. DOS is re-booted and your files are preserved. If we have 'auto init' not under your control, the files would have been wiped.....a case of being too clever!!

But I do not want to go through the process of 'INIT'-islising the RAMEX 128 every time I use the computer, I hear you say. Quite right, it can be done automatically using a simple "HELLO" program that can be incorporated in the 'Boot Strap'.

The DOS command you would use in your "HELLO" program is#INIT, Ss and this is the first of the new commands we have incorporated. These are fully explained and documented in chapter 6, but a simple "HELLO" program to carry out auto INIT incorporating this command, is given below. It can be later enhanced by using the #MOUNT command as explained in chapter 6.

Example HELLO program to INITialise RAMEX 128

```
 10  Rem RAMEX 128 INIT HELLO
 20  D$ = CHR$(4)
 30  HOME: VTAB10: HTAB4
 40  PRINT: "press RETURN to 'INIT' RAMEX 128"
 50  PRINT: HTAB7
 60  PRINT "or any other key to 'EXIT'"
 70  GET A$
 80  LET A = ASC(A$)
 90  IF A = 13 THEN GO TO 110
100  END
110  HOME
120  PRINT D$; "#INIT,Sn: Rem where n is the
                          RAMEX 128 slot No.
130  PRINT D$; "CATALOG"
140  END: Rem P.G.J. VERGECOURT 01.03.82.
```

See page 31 for further details on the command #INIT.

CHAPTER 6

Enhanced DOS commands for use with SOLIDOS

Introduction

This manual assumes that you are familiar with Apple DOS and the standard DOS commands available, as set out in the Apple DOS 3.3 manual, supplied with your Apple Disk Drive and controller.

DOS, or to give it its full name, Disk Operating System is the program loaded at system boot, from tracks 0, to 2 into the Apple. It is this program which has been amended to incorporate SOLIDOS with the extra commands, which are now listed and explained below. System programmers and those who are used to working in Machine Language, may find a detailed explanation of what we have done very helpful, and this is given in chapter 8 Please do not worry if that section is largely incomprehensible, because it is only intended for those programmers who may wish to amend DOS further and to use the RAMEX 128 successfully in its Disk Emulation Mode, you only need to understand the extra commands and their usage.

If you find it irritating that you don't understand chapter 8 and you want to delve further into the way DOS operates, may we suggest that you purchase a copy of "Beneath Apple DOS", by Don Worth and Pieter Lechner, which is published by Quality Software.

SECTION 1.

Outline of New DOS commands

Each new command is prefixed by the "✱" symbol, which differentiates it from a normal DOS command. Certain new commands and the enhanced commands have additional parameters "Zz" for "Z(!)lot z" and "Bb" for "Block/Bank b".

Even though some of the new commands are actually enhancements of Basic rather than of DOS, they have been made DOS commands for three reasons:-

(i) By combining them with DOS they are out of the way and do not affect page Ø or page 3, both of which are used by many standard programs or expansion cards.

(ii) It gives the maximum compatibility with most compilers. Because all commands in indirect mode are print statements, they are processed unaltered by the compiler. Of course, commands that look for line numbers or variables will not work.

(iii) Should a program containing these commands be run by error on a machine not containing SUPER HIDOS, you will get a tidy "Syntax Error" which will pinpoint the error immediately.

All the new commands can be used in direct or indirect mode. Parameters enclosed in square brackets are optional. If omitted, they take the value last used, with exception of the Block/Bank parameter B, whose default value is Ø.

New or enhanced DOS commands can be divided into four groups:-

(i) Commands used only in connection with SOLIDOS:

- ✱MOUNT
- ✱DUMP
- ✱PROTECT
- ✱CLEAR

(all of these use the parameter Z, either expressed or implied for the slot containing the SOLIDOS.

(ii) Commands used to access RAMEX cards:

- ✱CALL (only if Z parameter used)
- BLOAD "
- BRUN "
- BSAVE "

For these commands the parameter Z refers to the slot containing the card.

(iii) Commands usable with both SOLIDOS and ordinary Disk Drives:

- ✱INIT (slot parameter is essential)
- ✱STORE
- ✱RECALL
- ✱SAVE
- ✱ATTACH

(iv) Commands that are actually enhancements of Basic:

- ✱ERASE array
- ✱DEL xxx-yyy

Note

The comment 'Z'lot dependant during detailed discussion of each command is a warning that the command will have one of two affects dependant on which slot the RAMEX 128 is located.

SECTION 2.

The New 'House Keeping' commands

Command ✱INIT ,Ss [,Dd] [,Vv]

This command differs from the 'INIT' command in normal DOS, which of course is still available, in that it will 'INIT'ialise a RAMEX 128 or floppy Disk without saving the program in memory as the "HELLO" program.

This command or an ordinary 'INIT' must always be used in a program prior to any use of the RAMEX 128, to format and install the VTOC and directory. This command when executed, is performed almost instantaneously and will not slow the execution of a program to any degree.

It is this command which can be used in a bootstrap "HELLO" program, to 'INIT'ialise the RAMEX at System Boot, as shown in the sample program listing on page 27 of this manual.

This command can also be used within a program to 'wipe' clean a RAMEX 128, ready for a new set of Data or programs, though you would need to precede this command with a ✱DUMP (described later), if you wanted to save the contents to floppy prior to 'INIT'ialising.

The ✱INIT command also performs another useful function as follows: The command ✱MOUNT (described later), allows you to transfer the contents of a floppy Disk to a RAMEX 128 virtual Disk. Standard Disks would have been 'INIT'ialised using ordinary DOS via the command 'INIT HELLO' ,Ss, Dd etc.

If you now use the command ✱INIT on a blank or recirculated Disk and then transfer some Data to this new Disk from a RAMEX 128 with the new command ✱DUMP, you will find that the resultant Disk will "✱MOUNT" in only 20 - 25 seconds. An ordinary 'INIT'ialised Disk will still "MOUNT" although the transfer time is well over a minute so it may prove helpful to prepare new Disks, using this form of 'INIT'ialisation process to speed your loading time.

Command ✱MOUNT ,Zz ,Ss [,Dd] [,Vv] where z is the slot containing the RAMEX 128 and S is the slot No. of the Diskette to be mounted.

```
* * * * * * * * * * * * * * * * * * * *
* This command is 'Z'lot dependant. *
* * * * * * * * * * * * * * * * * * * *
```

WARNING

The command is designed to transfer in one operation all the Data tracks of a floppy Diskette to a RAMEX 128. There are two possible effects of this command, dependant on where your RAMEX 128 is located.

(1) RAMEX 128 in slot 1 - 6

In this configuration the command ✱MOUNT, will load and transfer tracks 3 thru 34, from the floppy Drive to the RAMEX 128, as nominated in the command syntax ,Zz ,Ss [,Dd][,Vv].

Tracks Ø thru 2 are the DOS tracks and are not required, as HIDOS is already resident in the system.

(ii) RAMEX 128 in slot Ø

In this configuration, HIDOS would be loaded automatically at system boot to Block 1 of the RAMEX 128, with the SOLIDOS enhancements and the DOS Interface loaded in ordinary Ram above relocated Himem, (see appendix C). If your "HELLO" program loaded Integer, then this would have been put onto Block Ø. Thus only 6 Blocks or 96K remain as virtual memory or virtual Disk.

Your floppy Disk has a total capacity of 140K for Data, and so not all this could 'fit' in the 96K remaining on the RAMEX 128, so the *MOUNT command limits the number of tracks transferred to 11 thru 34, from the floppy Diskette nominated in the syntax ,Zz ,Ss [,Dd][,Vv].

WARNING

```
* * * * * * * * * * * * * * * * * * * * * * * * * * *
* When 'MOUNT'ing a floppy Diskette to a RAMEX 128 in *
* slot Ø, it is possible to leave some Data/Programs  *
* behind as only tracks 11 thru 34 are transferred,   *
* but as the catalog track is transferred complete,   *
* then everything may appear to be resident with the  *
* VTOC showing space available on tracks 3 thru 10    *
* which do not even exist! A fail safe method is to   *
* initially transfer Data/Programs using "FID" and    *
* then '*DUMP' to a new *INIT'ialised floppy Disk.    *
* Any subsequent "MOUNT"s from this Disk will ensure  *
* a correct transfer of Data as only tracks 11 thru   *
* 34 will have been written to!                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * *
```


One of the obvious uses of the *MOUNT command would be in an applications program where the user would be prompted to insert his correct Data Disk, for transferral to the RAMEX 128 at the beginning of his session with the computer, or, a programmer doing development work, may wish all his program modules and subroutines transferred in a similar fashion with *MOUNT command contained in his Boot Strap program.

NB. See *INIT for details of "Speedy Mounts".

Command *DUMP ,Zz ,Ss [,Dd][,Vv].

WARNING

```
* * * * * * * * * * * * * * * * * * * * * * * * * * *
* This command is 'Z'lot dependant, but is also an   *
* unconditional Data Dump to the floppy Diskette nom- *
* inated. This command will overwrite all existing   *
* Data on a floppy, exclusive of the DOS Tracks. Care *
* must be taken that the correct Data/Program Disk is *
* installed to avoid irretrievable loss of Data.      *
* * * * * * * * * * * * * * * * * * * * * * * * * * *
```

This command is the complete reverse of the *MOUNT and the same restrictions apply when the RAMEX 128 is located in slot Ø. ie, Data from the RAMEX 128 would be saved back to a previously 'INIT'ialised floppy Diskette, starting at track 11 which was nominated under the syntax ,Zz ,Ss [,Dd][,Vv].

When the RAMEX is located in slot 1 - 6, then the Data is saved in tracks 3 thru 34 of the previously 'INIT' - ialised floppy Diskette nominated. An obvious use of this command would be when a user has selected "End of Session" from the menu of a applications program, he would be prompted to place a Data Disk in the floppy Drive, to have his day's work permanently stored.

Where A is the address of the program and where the value of 'A' may be in DECIMAL or HEX. Where Z is the 'Z'lot containing the RAMEX 128 and B is the Block and Bank addressed.

The value of B is calculated as shown in the command 'BSAVE'. If the parameter B is omitted, then the value will default to Block Ø, Bank 2 since DOS automatically resets B to zero each time it is accessed.

Two cases need considering with SUPER HIDOS.

(i) 'BRUN' a program in normal Ram.
When HIDOS is used to BRUN a program, it switches from DOS back to the current Basic before jumping to the starting address of the program. When a RTS is encountered at the end, control returns to DOS via the Interface, which switches DOS back on again. DOS then tidies up and passes control back to Basic. Thus in normal circumstances, Basic is on whilst a program is BRUN and DOS is inaccessible, without further switching or via routines on page 3.

This is not always convenient so provision has been made for DOS to be switched on whilst the program is BRUN.

This is achieved by a POKE 49145, 128

(ii) 'BRUN' a program on a RAMEX
Both DOS and Basic are obviously completely inaccessible under these conditions.

Selection of slot and Bank is made using the Z and B parameters exactly as for BLOAD and BSAVE.

Notes on Addresses

The jumps and routines at the top of page 3 are all available and may be used in the usual way. Thus if you are in Monitor, 3DØG will put you back into the current Basic exactly as normal.

To enable full use to be made of the routines on page 3 that give access to RWTS and the File MANAGER, it has been found necessary to relocate the IOB and the File MANAGER's parameter list on the Interface. (NB only when DOS is relocated to a RAMEX). Their starting addresses can be found by JSR $Ø3DC respectively, again as usual. The parameter list starts at $BFD2 and the IOB at $BFE8.

An additional facility is the transfer to page 3 of the starting address and length of a BLOAD'ed program. The length at $3E1, $3E2 and the starting address $B677, $B678, low byte first as normal.

If DOS is not relocated on to a RAMEX, then the normal start and end addresses of a 'BLOAD'ed Machine Code program are $0A00 lower than normal. Thus the start and length addresses normally found at $AA72, $AA73, $AA6Ø, $AA61 are now to be found at $AØ72, $AØ73, $AØ6Ø $AØ61 respectively and low byte first as normal.

Some information on the use of the RWTS routine is contained in "The DOS Manual", but for a detailed description and examples of use of both RWTS and the File MANAGER, reference should be made to "Beneath Apple DOS", by Worth and Lechner.

Command BSAVE <file name> ,A$xxxx,L$yyyy,Zz [,Bb]

Where Z is for 'Zlot' and z is a number from Ø to 7 and B is for the Block and Bank to be addressed. The value 'b' can be expressed in either HEX or DECIMAL, but if expressed as a DECIMAL, the value must not be a negative number. Please note the parameter Z is mandatory if the source of the program is on a RAMEX card. If Z alone is specified then the Block will be zero and the Bank 2 since DOS automatically resets B to zero each time it is accessed.

The value of 'b' is calculated as follows:

(i) Hexadecimal Notation

'b' = $[Block No] [Bank No]

Thus the value of 'b' for Block 5, Bank 2 would be $42.

A complete command would for a RAMEX in 'Z'lot 4 addressing Block 5, Bank 2 would be: BSAVE <file name> A$xxxx, L$yyyy, Z4,B$52.

(ii) Decimal Notation

'b' = [Block No. x 16] + Bank No.

Thus the value of 'b' for Block 5, Bank 2 would be 82. Similarly, the value of 'b' for Block 4, Bank 2 would be 66.

A complete command for a RAMEX in 'Z'lot 4 addressing Block 5, Bank 2 would be BSAVE <file name>, A$xxxx, L$yyyy, Z4,B82.

SECTION 4.

Numeric Array commands

Command ✻STORE <array name> [,Ss] [,Dd] [,Vv]

This command stores the Data portion of a numeric array to a floppy or virtual Disk under the name <array name> as nominated by the syntax [,Ss] [,Dd] [,Vv].

In Applesoft, only the first two characters (or first two + terminal % in an Integer array), are significant for an array, but the whole name is used by DOS. Thus :- Print D$; "STORE A3% file 1" would save the contents of an Integer array A3% under the name "A3/ file 1".

Similarly:- Print D$; "STORE AB"; JN$ with JN$= "XYZ LTD. (1982)", would save the contents of the array AB under the DOS file name "AB XYZ LTD (1982)".

The arrays are saved as Binary files and can be as little as half or a third the length of Data saved as text files.

Command ✻RECALL <array name> [,Ss] [,Dd] [,Vv]

This command RECALLS the Data portion of a numeric array that was put on Disk or virtual Disk with the command ✻SAVE.

Command ✻SAVE <file name> <xxxx[-yyyy>] [,Ss] [,Dd] [,Vv]
NB. use a "-" not a comma!

This command allows you to SAVE a program segment to floppy or virtual Disk in 'S'lots from line xxxx to line yyyy inclusive, under the name "file name".

This command is very useful for saving away sub-routines whilst carrying out program development, or for breaking a large program into segments, when constrained by the available memory in Motherboard Ram. These files can be retrieved by the command #ATTACH.

WARNING

```
* * * * * * * * * * * * * * * * * * * * * * * * * * *
* The line No. yyyy must be equal or greater to      *
* xxxx.  If yyyy is not specified, then it will      *
* default to 65534, ie. the end of a program.  These *
* segments are stored by #SAVE in a special way.  If *
* you require to look at these segments in isolation *
* then you must do the following or risk odd results.*
* Say your segment was #'SAVE'ed under the name of   *
* "TRIAL", then you must use the #ATTACH command as  *
* follows. #ATTACH TRIAL, Ø.  The program segment    *
* TRIAL can now be "LIST'ed, re-numbered "SAVE"ed etc.*
* in the normal way.                                 *
* * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Command #ATTACH <file name> <xxxx[-yyyy>] [,Ss] [,Dd][,Vv]

This command will load the program or programs segment called "file name" from Disk or virtual Disk and append it to the program already in memory with all variables saved between lines xxxx-1 and yyyy+1 and any existing lines between and including lines xxxx and yyyy will be overwritten.

If Programs #ATTACHED used strings, then may we suggest you use an ONERR GOTO routine to check if the "Program Too Large" error has been given. If so, then try a FRE(Ø), which should cure the problem if the program wasn't too large!

This command is obviously the reverse of #SAVE and allows you to retrieve programs or segments of programs 'SAVE'ed in this way, for inclusion in a program during "run time". (See warning in section on #SAVE).

SECTION 5.

Enhancements of Basic commands

Command #ERASE <array name> [,Ss] [,Dd] [,Vv]

This command removes from Disk or virtual Disk a numeric array that has been previously #'SAVE'ed.

WARNING

```
* * * * * * * * * * * * * * * * * * * * * * * * * * *
* The commands #STORE, #RECALL, #ERASE are only avail- *
* able for numeric arrays run with an Applesoft program *
* They will not work under Integer.                  *
* * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Command #DEL <xxxx [-yyyy>]
Note - not a comma!

This command can be used in either direct or indirect mode (ie. under program command), to delete lines in a program from xxxx-yyyy inclusive.

The parameter xxxx is mandatory, but if this is the only parameter specified, then all lines after and including xxxx to the end of the program will be deleted.

CHAPTER 7

Addressing the RAMEX 128

We apologise if this section is incomprehensible to the layman, but it is only intended as a guide to those machine language programmers who require more sophisticated detail, as to the RAMEX addresses used by the RAMEX 128. (Memory maps are given in Appendix B).

The RAMEX 128 is similar in operation to 16K Ram Cards, like the RAMEX 16, and uses its 128K of addressable memory in 16K Blocks. Thus in operation, each Block performs in a similar fashion to one 16K expansion card, which is why the RAMEX 128 can be used like a 16K Ram Card to run Pascal, 56K CP/M etc, (when placed in slot Ø).

Because the Apple I/O uses addresses $C000 through $CFFF, only the 12K space from $DØØØ to $FFFF is available to address each of the eight 16K Blocks available on the RAMEX 128.

Thus to address all of the 16K Block chosen, the lower 4K is used twice. You can decide under program command which 16K Block and which of the two lower 4K Banks you wish to occupy the space from $DØØØ to $DFFF, but only one 4K Bank at a time. This allows 8K of memory to be accessed in only 4K of address space.

The Apple on board ROMS, also share this memory space with the RAMEX 128. You may choose under program command, whether to Read ROM or RAMEX Ram, at memory addresses between $DØØØ to $FFFF, and whether you wish this area of Ram to be write-enabled or write-protected.

These functions are controlled by 'Control addresses', which we shall now discuss:-

The control addresses are used to control the following functions:-

(i) The write-enabling and protecting of the RAMEX 128 Ram.

(ii) RAMEX Read or On Board Rom Read.

(iii) The selection of which 4K Bank will be mapped into addresses $DØØØ to $DFFF, (the remaining Ram of each Block is directly addressable through $EØØØ to $FFFF).

(iv) To which of the eight 16K Blocks these instructions refer.

Block Select Addresses:-

Before setting the RAMEX 128 to read or write, the correct Block of 16K Ram must be chosen and this is done by accessing the following control addresses, and the LED indicators will light up in the following sequence, as a virtual check on your selection.

16K Block	Control Code	LED 4	5	6
Ø	COX 4	OFF	OFF	OFF
1	COX 5	ON	OFF	OFF
2	COX 6	OFF	ON	OFF
3	COX 7	ON	ON	OFF
4	COX C	OFF	OFF	ON
5	COX D	ON	OFF	ON
6	COX E	OFF	ON	ON
7	COF F	ON	ON	ON

(where X is the correct number for the slot selected, please dee Appendix A).

Block Control Addresses

The various control addresses and their functions are shown below for a RAMEX 128 Card fitted in slot Ø. For the control addresses of other slots, please refer to Appendix A.

	CONTROL ADDRESSES	FUNCTION SELECTED	4K Bank SELECTED
HEX DECIMAL	$C080 -16256	RAMEX Read and RAMEX Write protect	Bank 2
HEX DECIMAL	$C081 -16255	On Board Rom Read	Bank 2
HEX DECIMAL	$C082 -16254	On Board Rom Read and RAMEX Write protect	Bank 2
HEX DECIMAL	$C083 -16253	RAMEX Read	Bank 2
HEX DECIMAL	$C088 -16248	RAMEX Read and RAMEX Write protect	Bank 1
HEX DECIMAL	$C089 -16247	On Board Rom Read	Bank 1
HEX DECIMAL	$C08A -16246	On Board Rom Read and RAMEX Write protect	Bank 1
HEX DECIMAL	$C08B -16245	RAMEX Read	Bank 1

Please Note: On Board Rom refers to the Rom found on the Main Apple Motherboard.

If the following addresses are selected "Twice consecutively", then the RAMEX is also Write enabled, as follows:-

	CONTROL ADDRESS	FUNCTION SELECTED	4K Bank SELECTED
HEX DECIMAL	$C081 -16255	Write enables RAMEX	Bank 2
HEX DECIMAL	$C083 -16253	Write enables RAMEX	Bank 2
HEX DECIMAL	$C089 -16247	Write enables RAMEX	Bank 1
HEX DECIMAL	$C08B -16245	Write enables RAMEX	Bank 1

WARNING

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * *
* When changing between the 16K Blocks on the RAMEX   *
* 128, ensure that you Re-issue your correct control  *
* code to read or write etc., as any previous codes   *
* issued to another Block, will be altered in the     *
* move from one Block to another.                      *
* * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

At power on or by setting reset, (control reset on some computers), On Board Rom Read is automatically selected which disables RAMEX 128 Read. At the same time, the RAMEX is Write enabled with Bank 2, selected for mapping to $DØØØ to $DFFF, you will appreciate that this state is identical to that achieved by the double access of $C081 for a RAMEX 128 placed in slot Ø.

CHAPTER 8

How SUPER HIDOS works

1. The Interface

Why an Interface? Because the Basic in ROM, the relocated DOS, the virtual disk (SOLIDOS), and possibly another Basic on a firmware card or a RAMEX, are all occupying the same address range. The only way they can communicate with one another is via an "Interface" in ordinary RAM, which is accessible to all of them. (See Appendix B for details of the SUPER HIDOS Memory Map).

2. SUPER HIDOS Boot

(i) Bedtime Reading

It is assumed that the reader knows how a normal DOS boot takes place. If not, reference "Beneath Apple DOS" by Don Worth and Pieter Lechner, will make the following much clearer.

(ii) Master Boot

The first parts of the Boot (Boot Ø and Boot 1 in Worth and Lechner's terminology), are performed normally, but when Boot 2 is reached, there is a patch which transfers control to an extra routine. This first of all carries out Boot 2 and then loads the rest of track 2 (sectors 5 through 15) immediately above DOS at $4ØØØ to $4AFF. The program then returns to finish the normal Boot routine before jumping to $33BB (instead of $1BØ3 as normal).

At $33BB is the SUPER HIDOS Master program It first searches through the slots looking for RAMEX cards and makes a table of the results (initially at $4ACA, later moved to $BFCA), with a duplicate at $3FD2 for use during Slave Boots. There follows a decision procedure based on the Type number, as to whether DOS should stay in Motherboard RAM or be transferred to a RAMEX, and if so, to which one. The logic of this is described in detail in Chapter 2, Section 3.

The Master program now branches. If DOS is to be relocated on a RAMEX, all slot specific instructions are adjusted to suit the particular slot being used. The Monitor in ROM is then copied onto the RAMEX in that slot, after which the program proceeds to go through DOS making the necessary alterations. Most of the necessary alterations were made by the ENHANCER, but further ones are necessary to enable a relocated DOS to communicate with either Basic via the Interface.

This done, the two branches rejoin for the program to check whether ROM contains Integer or Floating Point and set flags accordingly. Control is then transferred back to the ordinary DOS Master program to relocate DOS and move the Interface. The final location of the Interface is from $B5ØØ to $BFFF for a relocated DOS and $B6ØØ to $BFFF for DOS in Motherboard RAM. A relocated DOS ends up at $D5ØØ to $F7FF on a RAMEX board or $93ØØ to $B5FF in Motherboard RAM.

(iii) The Slave Boot

When a Slave Diskette is 'INIT'ed, that part of the Interface containing the relocated IOB is saved to Track Ø, sectors A and B. This ensures that it will be available during Boot 2. Because of this, a Slave Boot is very much simpler than a Master Boot, since the Slave Diskette is preset for a particular arrangement of RAMEX board. A patch in Boot 1 on page 8 causes a jump to a routine on the same page that switches on the RAMEX card and copies the Monitor from ROM onto it. The range of the Boot is extended to include sectors A and B which are loaded direct to $BEØØ to $BFFF. After that, the Boot continues normally apart from another patch which loads the remainder of the Interface.

3. General

There are three DOS files formed at Boot as normal and page 3 of RAM is not required, making it available for the many programs that use it. The "HELLO" program put on the Master Diskette by the ENHANCER then checks to see if there is a RAMEX slot Ø available for Integer; if so, Integer Basic is then loaded.

WARNING

```
* * * * * * * * * * * * * * * * * * * * * * * * *
* The "HELLO" program on your Apple System Master  *
* Diskette will not work satisfactorily with a     *
* relocated DOS.                                   *
* * * * * * * * * * * * * * * * * * * * * * * * *
```


USEFUL TIPS

A.. Resetting the Micro

For those with autostart ROM, RESET functions in the normal way, returning you to whichever language you were in.

With the old monitor ROM, RESET may leave you with the soft switches set for DOS. The usual 3DØG will always take you back into your current BASIC, but if you attempt to use control C (or control B), the results will be unpredictable, if you are in fact set for DOS. While in monitor setting of the soft switches can always be checked by examining $EØØØ :

In Floating Point	$EØØØ = 4C
In Integer	$EØØØ = 2Ø
In DOS	$EØØØ = 91

(Because of the "RESET", SLTFLG may well be giving the wrong indication).

B. A Note on RAMEX switching

Although the soft switches on a RAMEX card are affected by any command which addresses them, some commands are more effective than others.

After a STA $CØ82
the double command
STA $CØ83
STA $CØ83
will enable you to read the RAMEX in slot Ø, but will leave it write protected.

E. ZFLAG ($BFFF)

Used with BLOAD, BSAVE and BRUN, if source or destination is on a RAMEX.

Value contained is (slot No.) * 16 Bank 2
(slot No.) * 16 + 8 for Bank 1

This can be loaded in registers X or Y and used as an Index for commands to switch the particular RAMEX to read, write etc., (e.g. STA C082, X etc.).

HIDOS ROUTINES

A. DOSON ($B628)

Makes the RAMEX card containing DOS (Bank 2) available for read and write. If entered from a RAMEX in slot Ø containing a Basic, switches it off first.
Does not affect any registers or flags but SLTFLG is reset.

B. BASIC ($B6ØØ)

Designed to be entered from DOS. Switches off DOS and returned to whichever Basic DOS is set for.
Does not affect any registers or flags but SLTFLG is reset.

C. CARDONI ($BE64)

Switches DOS off and another RAMEX or Bank on. Register X should contain (slot No.) * 16 + 8 if Bank 1, i.e. for slot 4, Bank 1, X = 72 ($48). Accumulator indeterminate. Flags N and Z may be affected. Does not alter SLTFLG.

APPENDIX A

LED Indicators

The LED's have been incorporated to enable a quick visual check of the status of your RAMEX 128. This is sometimes invaluable when debugging a complex machine code routine, involving a lot of Bank/Block switching. It is now possible to purchase through the Apple dealer network, a perspex lid (to replace your existing lid), which will enable you to see the LED's clearly whilst programming. Working with the lid off the computer, is not recommended by us or by Apple...BE WARNED!

LED 1	LED 3	4K Bank SELECTED	CONTROL CODE	
ON	OFF	2	COX 0)	
OFF	OFF	2	COX 1)	
OFF	OFF	2	COX 2)	where X in the
ON	OFF	2	COX 3)	control code
ON	ON	1	COX 8)	indicates
OFF	ON	1	COX 9)	RAMEX Slot.
OFF	ON	1	COX A)	
ON	ON	1	COX B)	

LED 4	LED 5	LED 6	16K Block SELECTED	CONTROL CODE	
OFF	OFF	OFF	Ø	COX 4)	
ON	OFF	OFF	1	COX 5)	
OFF	ON	OFF	2	COX 6)	where X in the
ON	ON	OFF	3	COX 7)	control code
OFF	OFF	ON	4	COX L)	indicates
ON	OFF	ON	5	COX D)	RAMEX Slot.
OFF	ON	ON	6	COX E)	
ON	ON	ON	7	COX F)	

In both tables, the value of 'X' in the control code indicates the slot in which the RAMEX 128 is located as shown below:-

SLOT	VALUE OF 'X'	SLOT	VALUE OF 'X'
Ø	8	4	C
1	9	5	D
2	A	6	E
3	B	7	F

If using more than one RAMEX 128, then remember to deselect one card, (COX 2 or COX A), before selecting another.

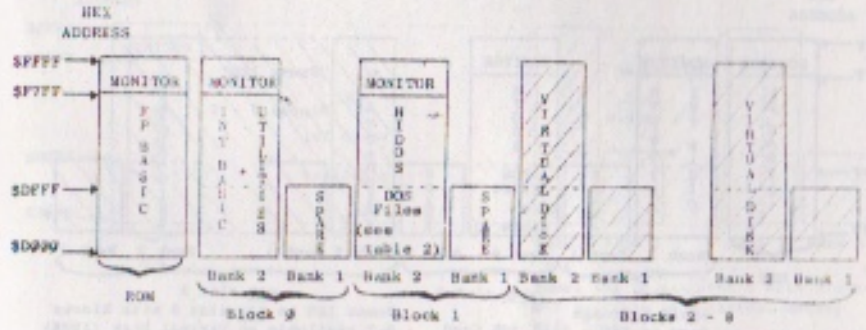
Note:

LED 2 is used to indicate the RAMEX 128 is in a Write enabled state and should be "on" at "System Power Up".

APPENDIX B

MEMORY MAPS

Configuration 1. Ramex 128 in slot Ø. DOS relocated in Block 1. Integer loaded to Block Ø. DOS Interface and DOS enhancements in normal Ram.



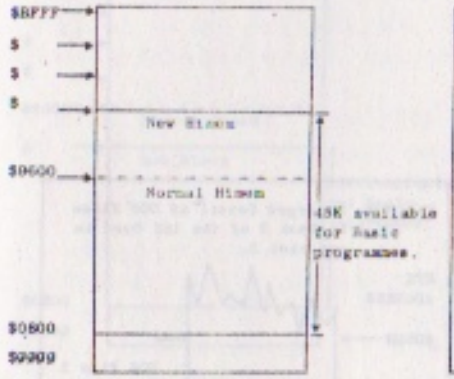
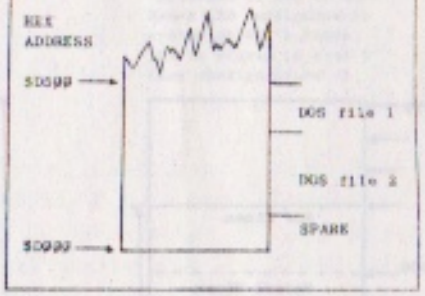
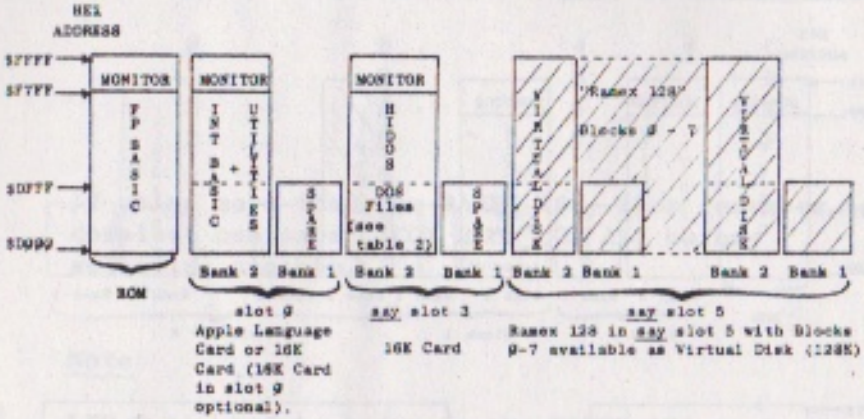


Table 2 Enlarged Detail of DOS Files in Block 1 of Ramex 128.



Configuration 2. Ramex 128 located in say slot 5 and Apple Language Card, (Ramex 16), in slot 0, with a second 16K Card (Ramex 16), in say slot 3.



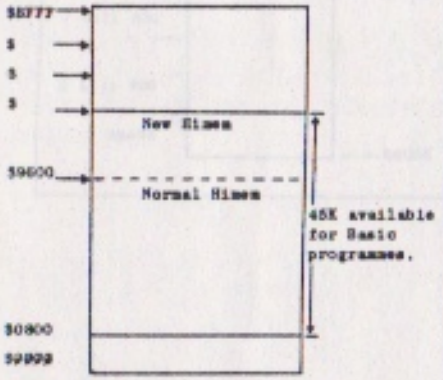
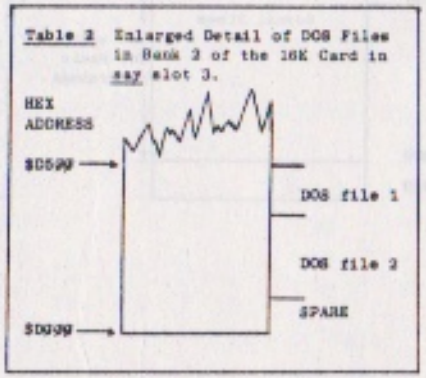
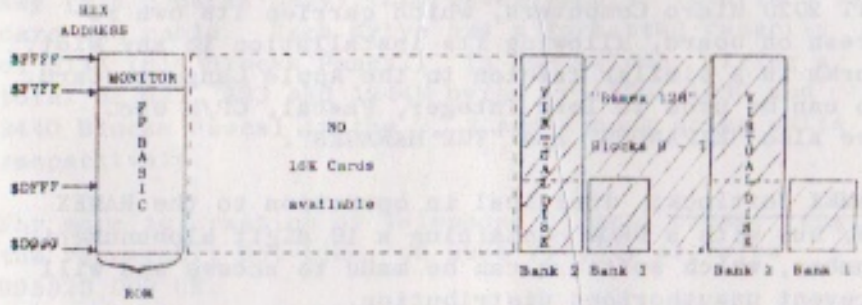


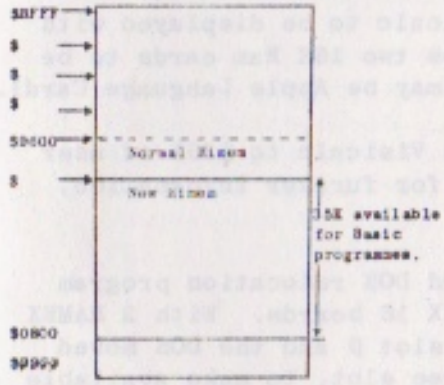
Table 2 Enlarged Detail of DOS Files in Bank 2 of the 16K Card in say slot 3.



Configuration 3. No Language Card or 16K Cards configured in system. Ramex 128 located in say slot 5 with DOS, DOS enhancements and interface in normal Ram.



Ramex 128 in say slot 5 with Blocks 0-7 available as a Virtual Memory Disk Drive.



N.B. Memory Map would remain unchanged if two or more Ramex 128 configured in system unless a Ramex 128 is placed in slot 0 (see configuration 1).

APPENDIX C

VERGECOURT PRODUCT RANGE

RAMEX 16: A 16K expansion board for the Apple II and ITT 2020 Micro Computers, which carries its own refresh on board, allowing its installation in any slot. Works in a similar fashion to the Apple Language Card, so can be used to load Integer, Pascal, CP/M etc., See also "EXPANDER" and "THE MANAGER".

RAMEX Softlock: Identical in operation to the RAMEX 16, but with a PROM containing a 10 digit alphanumeric number, which software can be made to access and will prevent unauthorised distribution.

Visicalc Expansion:

The EXPANDER: Enabling Visicalc to be displayed with 50K of user memory. Requires two 16K Ram cards to be resident in the system (one may be Apple Language Card).

The SUPER EXPANDER: Expands Visicalc to 136K of user memory. Requires RAMEX 128 for further information. See Chapter 4.

The MANAGER: A sophisticated DOS relocation program utilising either 1 or 2 RAMEX 16 boards. With 2 RAMEX Integer can be relocated in slot Ø and the DOS moved to the RAMEX in any other free slot, to make available 46K of Motherboard Ram. The utility programs on the DOS Systems Master, can also be updated to work with relocated DOS. The enhanced version of this program is supplied as standard on the 'ENHANCER' Diskette, supplied with the RAMEX 128.

Standard Interface Drive: Specifically designed to interface with the Apple II and ITT 2020 range of microcomputers, supplied in 2, 3 and 4 pack configurations, complete with case, power supply, security key (2 & 3 packs only), LED power supply, Interface card and cable. Each drive has a formatted capacity of 311K (610 Blocks Pascal), in one volume, giving a total of 622, 933 and 1244K bytes or 1220, 1830 and 2440 Blocks Pascal in the 2, 3 and 4 pack drive units respectively.

For more information or Telephone Orders, please ring the Vergecourt Hot-Line, on 0268-728484, or Telex 995323 DDP UK.

APPENDIX D

SERVICE INFORMATION

If your RAMEX requires repair, please return it to the dealer from whom it was purchased. If it is not possible to return the RAMEX to your dealer, you may send it directly to Vergecourt Ltd.

If repair is required during the warranty period, please enclose proof of purchase. During warranty, we will replace or repair your RAMEX 128 without charge.

If the RAMEX 128 requires service after the warranty period expires, it will be repaired for a flat fee of £75.00. The service charge does not cover damage due to negligence, misuse, or inadequate packaging on return to Vergecourt Ltd.

To return your RAMEX 128 for service, please mail it post-paid to Vergecourt Ltd. Package the card securely and we advise using the original packaging, as we cannot be responsible for damage due to shipping. BE SURE to enclose proof of purchase for warranty work or a cheque or money order for £75.00 + VAT, for non-warranty repairs.

Mail post-paid to:

Vergecourt Ltd.,
17, Nobel Square,
BASILDON,
Essex.
SS13 1LP.

APPENDIX E

Pascal Virtual Disk

On the back of your Enhancer Disk is a Pascal program that will create a Virtual or Pseudo Pascal Disk Drive.

The Disk contains both Text and Code files. The Driver is written in Assembler, and the "SYSTEM.STARTUP" program which formats the RAMEX Pseudo - Directory, is written in Pascal.

The "SYSTEM.ATTACH" utility installs the Driver in the System upon Boot and assumes device #12.

The program is well documented and can be modified by the user to suit his individual needs.