ADS

Integrater

Operation & Installation

Manual

For the Corvus and RAMDISK

Forward

Computer Manuals! Who needs 'em?

Yes, we too have dredged our way through manuals filled with sludge, in search of how to use a particular piece of hardware or software. Deciphering computer documentation masquerading as an instruction manual is one of the horrors of the computer industry.

So we have tried something a bit different in this manual. You will find the style tutorial and friendly, with examples to try, and not very much abstract theory. As much as possible, we have tried to avoid the computer shorthand, the dreaded "Jargon" that prevents so many computer users from using their machines.

We cannot cover every application in this manual in absolute detail, for RAMDISKs and Corvuses are being used in widely varying applications, in ways we can't even imagine. However, we feel that if we give you the tools and some basic tutoring on how to use them, you will see how they can be useful for your particular application. The Integrater is an extremely powerful tool, designed with the same powerful philosophy as the Atari: let the user control as much as they want to, from very little (for a beginner) to all (for an expert). We have tried to make the Integrater as easy to use as possible for a beginner, yet as powerful as possible for someone who wishes to really use their system to its limits.

Enjoy your Integrater board. There is nothing else like it on the market. It has been needed for some time, and the need is now filled.

Introduction

Congratulations on purchasing the Integrater card for your Atari computer. We believe it is the finest add-on product for those Atari users with high performance mass storage devices, such as the RAMDISK or Corvus. For the first time these devices will function as if Atari had designed them into the product line and given them full support in Atari's operating system software.

For years the RAMDISK has supplied extremely high speed disk "emulation" for Atari users. However, the RAMDISK had to be run with a patched operating system, the Axlon MMS. Software which was incompatible with this operating system (for instance, nearly anything copy protected or anything with its own disk operating system) could not use the power of the RAMDISK.

The Corvus mass storage Winchester disk drive also has proven itself with the Atari community. But again, a patched operating system had to be used; the Corvus was not compatible with copy protected software nor with any software requiring a different DOS.

The situation has been getting worse for RAMDISK and Corvus owners because an increasing percentage of software for the Atari comes in copy-protected formnat and cannot be used with either product. This trend is continuing to accelerate. Hence less and less software is available that can be used with the RAMDISK or Corvus.

Enter the Integrater. The Integrater began when a Allan Adams, a local Atari user contacted David & Sandy Small (DSSA) about using Atari's Microsoft Basic with the Corvus. Microsoft BASIC loads into the same area of memory which the Corvus uses as part of its operating system and is thus incompatible with the Corvus. However, with certain modifications to the Atari's operating system, we found that the Corvus could be made compatible with Microsoft Basic, and, in fact, with any other application that uses disk storage. We extended the idea to include the RAMDISK, which had essentially the same compatibility problems as the Corvus, and came up with the Integrater board. The result (in hardware) was the Integrater board; in software, smallDOS, a custom Disk Operating System for the RAMDISK and Corvus.

It should be noted that when Atari designed the 800 machine, the "10K ROM" board was designated a "Personality Module". At that time Atari planned to design several different "Personality Modules" for the 800 to help it to perform different tasks. Unfortunately, this never came to pass, so there's only one personality for the 800 -- at least from Atari. We have just created a second personality for the 800, for users with RAMDISKs and Corvuses. This is consistent with the philosophy of the people who designed the 800, not just an "addon" or "patch" board.

Hence, what you have bought is a new personality board for the Atari. This new personality (operating system) is the same as the old; however, it adds some new features. This new personality gets along quite well with the Corvus and RAMDISK, and the old one did not. We chose the name "Integrater" because the new personality module "integrates" the Corvus and RAMDISK into the entire Atari electrical and software structure seamlessly, without the loose ends that characterized previous attempts to merge the two.

There's little doubt that there are more applications for the Integrater; we are working on more in-house. If you have a mass storage, printer, or other operating system need that the Integrater could solve (for instance, running 8 inch disks or hard disks with the Atari) feel free to give us a call; there might be an Integrater just for you.

How the Integrater Works
(for a beginner)

The Integrater functions by waiting for a disk access request to occur inside the Atari. A "disk acccess request" occurs when some software inside the Atari needs to do something to the disk, for instance, to read or write data, or perhaps format a blank diskette. When a request for a read or write to the disk is made, the Integrater switches on, and redirects the disk request to either the Corvus, RAMDISK, or the Atari disk drive. All of this redirection occurs within the Integrater.

This is the basic idea behind The Integrater: redirecting disk requests to the proper place without the user having to worry about special operating systems, patches, and the like.

The Atari uses the "Serial Bus" cable to connect to its disk drives. This serial bus is the cable that connects from the side of your Atari to the back of the disk drive. Inside the Atari is a software routine called "SIO", for "Serial Input - Output", which runs this serial cable. When a disk access request is made in your software, ultimately it ends up at SIO, because (normally) all such requests go out over the disk serial cable.

The Integrater intercepts all requests which reach SIO and decides if the request should be sent (via SIO) to an Atari 810 disk drive. The other possibilities are out the front ports, to a possible Corvus, or internally to a possible RAMDISK. Should the Integrater decide to let the request reach the serial I/O routine, then the request will be routed to the Atari 810 disk without any further handling. This is important because it means that the Integrater board is fully compatible with all software that uses the disk drive; if the Integrater decides that the disk request should go to the 810 disk drive, it sends the request there, undetectably. The user and the software never knows that the Integrater intercepted the disk request and then allowed it to proceed to SIO.

On the other hand, if a RAMDISK or Corvus requrest is made, the Integrater then routes the disk request to those devices and does not allow the request to reach SIO.

Furthermore, this switching capability is at your command at any time. You control what happens to disk data requests and can change your mind at any time. This allows you power over your computer's mass storage which has never existed before for the Atari.

The Corvus and RAMDISK are treated as 810 - like devices, and the Atari doesn't "see" any difference between them and the 810. The Corvus and RAMDISK are made fully compatible and equal in the computer's eye. Hence these devices become compatible with the software they could not be used with before, such as copy protected software or software with a custom "Disk Operating System", or DOS. A few examples might be FORTH, K-DOS, or Atari's new DOS 3.0. All of your disk utilities will now work with the RAMDISK and Corvus. All of your word processors, spreadsheets, and other programs can now use the mass storage of the Corvus or the speed of the RAMDISK.

Remember that the Integrater is far more than just a way to make the Atari talk to the RAMDISK or Corvus without the compatability hassles. It adds complete new features to the Atari's personality. Along these lines, we have added some features to the Integrater that we feel you will find most welcome:

* The RAMDISK now runs even faster, about 2 times faster than with the Axlon MMS RAMDISK software.

* The RAMDISK can be designated as any drive without rebooting the Atari or using an applications program.

* The user can now "boot", or start up, the Atari directly from the Corvus, without any 810 disk drive at all. First, this saves the cost of a $500 disk drive. Second, it saves the inconvenience and unreliability of resorting to disk or tape as a startup procedure. In networked systems of many Ataris, the Integrater can save a great deal of money and add great power to the Corvus network.

* The Corvus can now "boot" directly into an applications program using standard Atari AUTORUN.SYS techniques and programs.

* The slow and inconvenient Corvus "Mount" program has been rewritten in assembly language, which runs at high speed, and this new program can be called from any software under user control. This means you can read or write to any portion of the Corvus and don't have to stop to run a Corvus "Mount" program first. In addition, the Mount Table Editor (as it is called) gives you, for the first time, powerful control over your Atari's disk system.

* All the Integrater's software for redirecting disk accesses (known as smallDOS), including everything displayed by the Mount Table Editor, is in a section of memory the Atari doesn't use for anything, Great care has been taken to ensure compatability with aftermarket plug-ins for the Atari, such as the BIT-3 80-column board, the California Microlink Parallel I/O board, the Austin Franklin RGB board, and so forth. In tetrms of memory usage, you will notice no change; the Integrater uses none of the Atari's memory, so you don't need to worry about "running out of space". Indeed, Corvus users are in for an agreeable surprise, because:

* Corvus users gain about 1K of free memory by using a normal DOS (like Atari's DOS 2.0S) over the Corvus DOS.

* We have made great effort to make this product truly useful across a wide spectrum of possible applications. For instance, some programs allow only two disk drives (D1: and D2:) to be accessed, so we made the RAMDISK "default" to being drive 2. This way, nearly any program can use the RAMDISK as drive 2. If we had made it drive 8, for instance, many programs would not have been able to use the RAMDISK, because they do not allow access to drive 8. You will find many other design considerations were based off of making a usable product vs. making something totally new which was incompatible with software already on the market. To sum up, we have tried very hard to keep our product compatible with existing software, with anyone's DOS, and with anyone's copy protection. We want this to be a product you can use with any software with no problems.

Compatibility is the name of the game. Too many times we have seen hardware products which expect software manufacturers to make modifications to their products to work with the hardware product. What happens is that the software manufacturer hasn't the time or interest to do so, and the customer is left stuck with a piece of hardware that none of his software works with. The Integrater cures two such problems, those of the Corvus and RAMDISK.

We should mention our competition. There is none for the RAMDISK market. For the Corvus market, there is a cartridge (for the right hand slot) which boots the Atari from the Corvus. However, it gives you none of the powerful abilities of our Mount Table Editor nor does it give you compatibility across different software. We have easily the best interface for these products for the market, and at a low cost as well.

Let's get on with installing the Integrater so that you can take advantage of all of this power.

Installation

If you have ever had to wire in a new piece of equipment to a computer, or make a cable to connect a modem, or any of these sorts of things, forget your worries. Installing the Integrater is a snap. That's why this section is so short. Remember, we made this thing to be usable.

1. Open the Atari's lid the way you would to replace a cartridge (like BASIC or Pac-Man).

2. There are two little hold-down clamps now visible on either side of the Atari's main upper lid. Flip them so they no longer hold the main lid down. NOTE: On new Atari's, these have been replaced with a simple screw. If this is the case, just remove the screws.

3. Remove the Atari's top lid. It lifts up and towards you. Don't lean on it too hard; a bit of firm pressure should be adequate to pull it up and towards you.

4. Looking at the Atari, there are from two to four boards now revealed, plugged into sockets at the base of the Atari. The boards are labelled "16K RAM" or the like. The board closest to you is marked "10K-ROM". It needs to be removed.

A. Try grasping it by its outer handles and pulling up. On most Ataris, it will come out easily.

B. If it is in there pretty tight, take a screwdriver and GENTLY pry the board upwards, by placing the middle of the screwdriver on the aluminum case surrounding all the boards and the tip on the ridges of the board's handles. Pry up a little at a time until the thing loosens up. Pliers also work. Again, if you do pry, pry just a little bit, on one side, and then the other; don't try to pop the board out all from one side. Once you loosen the board, it ought to come out in your hand.

C. Don't even think about using WD-40 or other solvent, lubricating sprays.

D. Another good approach is to pull out the board just behind the "10K-ROM" board, put your hand along the full length of the "10K-ROM" board, and with the added friction, helping pull up.

5. Great, you have the board out. Now, did you order the Integrater WITH "B" revision operating system chips (i.e., the more expensive option?) If so, you don't need to pull the chips out of your old O.S. board (the "O.S." board is the same as the "10K-ROM" board), so skip to step 7, installation.

6. Since you ordered your Integrater WITHOUT the O.S. chips, we need to pull them out of your old board and install them into the

Integrater. This is not difficult, but you do need to be careful.

A. Get a phillips head screwdriver and remove the metal cover to the "10K-ROM" board. There are two screws. Once you take them out, the metal cover will fall into your hand when you turn the board upside down and tap gently.

B. The cover will now "open" along the base, where the shiny contact pins are. This will take just a tad of persuasion; the board is very lightly glued. The label at the top ("10K-ROM" serves as the hinge, by the way. Open the plastic cover up slightly and pull out the circuit board within. Don't worry about being shocked or anything; there is no electricity in the board. You'll end up with an empty plastic case and a circuit board in your hand.

C. Hold the board with the chips facing up and the row of contact fingers (silver or gold) facing you.

D. The three big chips (with 24 pins, if you wish to count them) on the leftmost, next to leftmost, and next to that need to come out. We'll get to that in a moment; we just wanted to show you the chips we're speaking of.

E. Unpack the Integrater board. There are three empty sockets where these three chips are going to go. Now here comes the confusing part: the chips move to a new position on the nerw board. (We had to do it this way for design reasons). On the old board, the chips are installed like this:

```
 F    E    D
chip chip chip
```

On the new board, they go like this:

```
(existing chip)  (ADS R83 chip)     D    E    F
                                  chip chip chip
```

Note that the order is reversed. The leftmost chip on your old O.S. board becomes the rightmost chip on the Integrater board.

We STRONGLY recommend you swap the chips one at time. That way, there is no possibility of a mixup. It won't hurt anything if you get two of them mixed up, in terms of the Atari (so don't get any mental pictures of smoke and flames), the Atari just won't function. So please do it one at a time.

F. Using a flatblade screwdriver or durable knife, pry the leftmost (F) chip up out out of its socket., The best way is to get the screwdriver just started under the chip, then slide the screwdriver the full length under the chip. Try not to lever the chip out at angle, as that bends the pins at the base of the chip. We recommend you work from the top of the board; slide the screwdriver under the chip, and pick it up.

Remember the chip is in the socket. Don't try to pry the socket off the board, just the chip.

When you get the chip out, make sure you didn't bend any pins so badly that they can't be reinstalled. If they are bent, a little judicious use of needle-nosed pliers can work wonders. Note that these pins can only be bent back and forth a few times before breaking off, so be careful.

Take the chip, and with the little notch or hole cut in one end pointing up, put it into the rightmost socket of the Integrater. Be sure to check that the notch is up, pointing towards the top of the board and next to the top. This chip should go in the rightmost socket. (The serial number on the chip should end with the numbers "599" -- this is the "F" chip).

Won't go in? Make sure all the pins are started into their holes, and one isn't bending in or out. A little firm force is necessary sometimes. Remember, even the pros sometimes bend a pin or get a "leg up" when installing a chip, so stay after it. One good tehcnique I use is to start all the pins on one side, press the chip towards that side until the pins on the other side match with their holes, and "gently but firmly" put the chip in the socket.

Next, check your work by sighting up the legs of the chips from the board's edge. Look down the board, with it at eye level, and you'll be able to se any legs that are bent in, out, or out of shape. If you find one, pop the chip out, straighten the leg with pliers, and be more careful next time. If you break a leg or chip, call -- we'll arrange something.

Next, remove the middle chip from your old O.S.card, with the "499" in its serial number. This is the "E" chip. It goes in the middle socket of the Integrater board, to the left of the one just installed. Again, be sure to check the installation.

Finally, do the rightmost chip in your old O.S. board, with the "399" in its serial number. Pull it out of your old O.S. and install it into the one empty socket of the Integrater. This is the "D" chip.

Okay, all done. One final check: the chips, going left to right, on the Integrater, should be:

```
(existing RAM chip) (ADS chip) 399 chip / 499 chip / 599 chip.
                                "D"         "E"         "F"
```

Got it? Good; let's install the board now.

7. Plug the Integrater board into the Atari where the old "10K-ROM" board was. (We do not recommend putting the plastic case on this board; it doesn't fit.) The chips are facing away from you as you plug it in. It will plug in SOLIDLY and be firm; make sure its seated. Go ahead, put a little pressure on it and make sure it is all the way down.

Believe me, you'll have nothing but grief if the board isn't seated or if it openly wobbles. The contact fingers are extremely sensitive to vibration and grime, so make sure the board is in and seated. Most likely you will be able to pick the Atari up by the Integrater board, it will be in there so solidly. That's good. (Be careful not to drop the machine if you try this and the board comes loose!)

Some troubles that many Atari users have had with their machines can be directly traced to contact problems with the RAM and ROM boards, by the way, so all this caution is really needed.

If the board seems to wobble despite being in there solidly, put a piece of cotton in between it and the aluminum case (in the case) and it and the first RAM board (front), to help cut down the wobble.

Okay, all done. Plug in any RAM boards you might have removed during the installation process, and we're finished.

8. Replace the cover. This is surprisingly hard for many. Here's the key: look at the top of the cover, in the back. There are two small metal tabs sticking out. These match to two small slots in the back of the Atari. Match those two slots up with the tabs by pressing down on the back of the cover as you try to install it, and they'll slide right in. The cover goes down and away from you as you install it. Next, put either the screws or the small tabs back and in place, and you're done.

9. The Moment of Truth. Close the cartridge door, unplug the "serial bus" peripheral cable, and turn it on.

You should immediately get a blue screen, a 1-2 second delay, then "Atari Memo Pad". If you get whirling colors, a flat black screen, a "beep", or anything else, the board isn't right. Open up the Atari and check:

1. Is the board in solidly? Are the chips all pointing towards the back?

2. Are all the other boards back in?

3. Are the chips you may have installed in in the right sequence, left to right (399/499/599)?

4. Do all the chips have the "notch" pointing up?

5. Is there anything obviously wrong? A wire torn loose on the jumpering or anything? If so, call us.

Assuming you get the blue screen, we're nearly there. Next, turn the Atari off, wait two seconds, press the OPTION key down, and turn it on. You should be presented with a green screen with the legend, "MOUNT TABLE EDITOR" and all sorts of text and tables. If you get this far, things are most likely okay.

By the way, get in the habit of turning your Atari off, waiting two seconds, then turning it on. First, its better for the Atari's power supply. Second, it gives our board a chance to re-initialize to all the default conditions; if you flip the switch off-on, then you might get some strange results from part of our tables being scrozzled. When the board realizes that you've powered off (by its memory going away), it resets everything, but give it a chance -- the big capacitors in the Atari hold the power up to all the chips for a second or so after you turn the power switch off.

Okay, we're all done. Where to from here? If you're going to be using the Integrater with a RAMDISK, read the "Disk Mapping" section that follows, and then the RAMDISK section. If you're going to be using the Corvus, read the "Disk Mapping" section that follows, then the Corvus section. Finally, in either case, read about the Mount Table Editor. If you have both, then read everything.

Good luck!

Disk Mapping

It would be best right now to introduce you to the concept of disk mapping, because it underlies everything we'll be talking about with the Corvus or RAMDISK.

The Atari 800 was set up, originally, to use eight disk drives. The manuals all say four, because after the design was completed Atari dropped the 815 disk drive, which went up to #8. However, the software was already written, so 1-8 it stayed. (Regrettably, some software forces you into 1-4 because the manuals are written that way.)

When you make a disk access, the type of request and the drive number are normally shipped down the serial bus cable by SIO, the Serial Input - Output software. SIO is called up anytime you do anything to the disk drive from software. Well, I'd like you to consider that the Atari doesn't know if a disk drive is connected or what is connected to that serial bus when it makes the request; it sends the request, and waits hopefully for something hooked on that bus to do something with it. If not, you get an ERROR-138: Device Timeout (the Atari got tired of waiting for something to happen).

But the Atari really doesn't care who responds to the request, as long as someone does. For instance, if you have a Percom drive, the drive will work equally well with the Atari. It does the same thing the Atari drive does: watch the serial bus for requests sent to it, and answer them.

Now consider the D1-D8 as not having to be disk drives. They are "windows" to the outside world. When you send a request to D2:, stop thinking of it as going to disk #2 -- think of it as going to window #2 or port #2, and to whatever device is hooked up to that particular window. For instance, if I have a Percom disk drive hooked up to window #1, a Corvus to window #2, and a RAMDISK to window #3, then if I access D1:, I will get the Percom. If I access D3:, I will get the RAMDISK. And if I access D2:, I will get the Corvus.

Get the idea? Just because it says "D" as the device name doesn't mean the request HAS to go to an Atari disk drive. It just means the request goes out to a given window number from one to eight. Whatever is connected to that window will respond to the disk request.

The Mount Table Editor, part of the Integrater's smallDOS software, gives you absolute control over these windows. Before, whenever a request was made to window #1, the Atari forced it to disk drive #1. Now the window is under your control. You can direct requests to window #1 to Atari drive 2, for instance, or to the Corvus or RAMDISK.

Keep in mind that D1-D8 do NOT mean specific disk drive numbers when used from the Atari end. They just mean "Access window #n and give it this disk request". Whatever is connected to that window will then respond to the disk request, and the Atari will happily go on its way thinking it was talking to a disk.

The Integrater uses a "Mount Table" to tell it which window # goes to what device. This Mount Table is set up as follows on powerup:

1. All Dn:'s are set to Atari drive #'s 1-8.
   i.e., D1: goes to Atari drive #1.

2. Is a Corvus online? If so, read in its mount table and use it. This would override #1.

3. Is a RAMDISK online? If so, make it D2:. This will override both 1 and 2. Also, initialize the RAMDISK to look like a blank, formatted diskette. Note: This D2: will override the Corvus mount table D2:.

4. Proceed to start the system.

Hence if you do not connect a Corvus or RAMDISK, you will have the standard Atari setup. If you connect a RAMDISK only, you will have the standard Atari setup with D2: sent to the RAMDISK. If you have a Corvus, use its Mount Table.

This might all sound complex, but what it all boils down to is that everything is taken care of for you as much as possible with defaults. If you don't know what you're doing with the Integrater, and haven't time to learn to properly use it, then there are certain things you can count on happening. As we said, we have tried to make this software as user friendly as possible.

Next, please read either the "RAMDISK" or "CORVUS" sections (or possibly both), depending on which device you have.

RAMDISK Operation

Right now, if you own a RAMDISK, you must use the Axlon MMS as the Disk Operating System. No more! With the Integrater, your RAMDISK gets along with any D.O.S. (with an exception noted below).

Briefly, the RAMDISK may be assigned to any of the 1-8 windows and will respond as a very fast disk drive to a request made to that window. For instance, if you have a RAMDISK installed, it will default to window #2. Thus any requests to window #2 (D2:) will go to the RAMDISK. Try this example:

1. Plug in BASIC.

2. Make sure your RAMDISK is on (if it has the switch).

3. Turn the system on.
   (Note: at powerup, when the RAMDISK is switched on, the RAMDISK is automatically initialized to "look" like a blank, formatted diskette. You don't need to format the RAMDISK first before trying to use it. We did this because some application programs do not have a "format" option.)

4. Wait until you get a ready prompt. Next, type in a short program:

```
10 FOR A=1 to 10
20 PRINT "I AM GOING TO BE SAVED"
30 NEXT A
40 END
```

and then type

```
SAVE"D2:PROG"
```

Almost instantly the program will be saved to the RAMDISK. An Atari disk drive hooked up as #2 won't be turned on or used, the RAMDISK will. This is because the RAMDISK is assigned to window #2 and is accesses whenever you use "D2:".

Okay, we have the program saved to the RAMDISK. Type

```
NEW
```

to clear out the program space.

Type

```
LIST
```

and you'll see that no program is loaded. Next, let's fetch the program back off the RAMDISK:

```
LOAD"D2:PROG"
```

and zap (very quickly) your program will load in. A

```
LIST
```

will confirm it has indeed loaded.

If you loaded DOS from disk, you will now be able to go to DOS and get a directory of D2:. There you will find your program PROG, just as if you were using a regular disk drive.

Next, boot up the system with an Atari DOS disk in drive #1. Go to the DOS menu by typing DOS. If you do a directory of D2:, you will get a directory of what's attached to window #2, or the RAMDISK. It will say, "707 Free Sectors"-- in other words, the same thing a blank, formatted disk would say. On powerup, the RAMDISK looks just like a formatted, blank diskette, with the same number of sectors and a blank directory.

We can write DOS files to the RAMDISK to demonstrate how speedy it is. Select option H, tell it Drive #2 (which will go to Window 2 and thus to the RAMDISK), and "Y" to confirm writing, and in nothing flat the DOS files will be written to the RAMDISK. Do a directory of drive #2 ("A", return, "D2:", return will do it) and you'll see the DOS files written on the RAMDISK.

Incidentally, you could not have written the dos files to the RAMDISK under the old operating system (Axlon's MMS). Just another nice touch on our part.

Okay, erase everything on drive 2. Do "D", then "D2:*.*/N" to erase the DOS files. In nothing flat, the system will be finished.

Format the RAMDISK? Okay, use the I option to do it. Copy files to it? No problem. Duplicate disk? Again, it works. The point I'm trying to get across is that the RAMDISK behaves exactly like a disk drive running at ultrahigh speed inside your Atari, and the Integrater board handles all the problems with getting requests to "D2:" to reach the RAMDISK.

Let's say we are in the middle of a popular game, Zork. We ask to save the game's current state so we can come back another day to play it. The machine asks us which disk drive to save data to; we tell it #2. What will happen? Zork will save the game state to the RAMDISK, at impressive speed. There won't be compatibility problems of any sort.

Or, perhaps, we are in the middle of FileManager 800 or other databases. We want to do a "sort" on data. If the data needs to come from the disk drive, it will take forever, but off the RAMDISK, it will be very fast. So we use Filemanager to copy the data desired from diskette to D2: (alias the RAMDISK), and do our operations from D2:. Everything will happen very quickly because of the RAMDISK, and again, the smallDOS in the Integrater will worry about all the compatibility problems.

If you want to use disk utilities on the data stored in the RAMDISK, feel free. Sector copiers also work. (Neither of these would work with the old system -- again, compatability, compatability).

There are a few notes about the operation for advanced users:

The RAMDISK works by taking the area of memory from $4000-$7FFF and making 8 banks of 16K available. Normally, you run with bank #7 -- all the screen and programs and stuff that normally would be in at 4000-7FFF are in bank 7. Banks 0-5 are used for disk emulation, two sectors per page of memory.

Note that if your program makes use of player-missile graphics or a graphics memory in the 4000-7FFF areas, there will be a slight flicker as memory is paged in and out during the access to the RAMDISK.

The RAMDISK is paged by writing a number into memory anywhere from $0FF0-$0FFF or $CFF0-$CFFF. This has caused a number of problems with users. Many programs write to $FF0-$FFF during execution as part of their working and flip the RAMDISK in and out randomly. The Integrater cannot cure this; it helps by resetting the RAMDISK to page 7 whenever a disk access occurs, but there's only so much it can do. This is just a tradeoff in the RAMDISK, and really should have been fixed at the factory, but wasn't. If a given program absolutely will not work with the RAMDISK, don't be quick to blame the Integrater -- try the program without the Integrater installed and see if the RAMDISK isn't getting mangled. If there is enough interest we can provide a "fix" for this problem, by forcing only accesses to $CFF0 to affect the RAMDISK; the $C000 4K area is unused.

Note that the RAMDISK runs about twice as fast as it did under the old software. We made the essential transfer loop very efficient, and the speedup is the result of this.

That about wraps up the RAMDISK discussion. We're sure you will find the RAMDISK a much more powerful, more integrated part of the Atari operating system with your Integrater board.

Corvus Operation

The first step to using the Integrater with the Corvus is to create a working Corvus system using the present Corvus Disk Operating System, DOS A2.0D. This can be done by following the directions in the Corvus manual and using the supplied Corvus software. In particular, the Corvus disk must be initialized, the number of single and double density volumes determined, and so forth. Why is this necessary? The Integrater uses these tables to function.

Also, this manual cannot cover the operation of the Corvus in detail; we must assume some knowledge about the Corvus. This knowledge can be obtained from the Corvus manuals. In particular, we suggest experimenting with the Corvus program "Mount", a precursor to the Mount Table Editor found in the Integrater. You should be familiar with the concept of volumes and mount/dismounting them.

With the Corvus configured, we can now proceed with installation of the Integrater software support.

Let's assume you have the Corvus configured as it is when first initialized; Corvus volumes 1-7 are attached to D1:-D7:, and D8: is the Atari disk drive #1. (If you have altered this configuration via the Corvus Mount program, please change it back).

Our first step is to install a working Disk Operating System with "boot sectors" onto Corvus volume #1. This is needed because the Atari, on powerup, reads in "boot sectors" from whatever disk is connected to window #1. The present Corvus system has no boot sectors, as a different scheme is used to start up the machine (an Atari 810). Here's the procedure (Note: do this without the Integrater board installed):

1. Select a blank Corvus volume the same size (single or double density) as volume #1.

2. From the A2.0D (Corvus DOS) Menu, ensure that the volume you have selected is blank. For now, let's assume you will use volume #2. One quick method is to reformat Volume 2 with I,2,Y.

3. Use "C" to copy files, then enter "D1:*.*,D2:*.*" (no quotation marks) as the copy information. All the Corvus files from Volume #1 (such as Mount, Mirror, and so forth) will be copied from Volume 1 to Volume 2.

Okay, we now have a copy of the information on volume 1 stored for later reference. This includes the Corvus utilities such as Mirror, Mount, and Diag. Next, let's put a new DOS on the Corvus.

4. Put a disk with Atari DOS 2.0S or an equivalent into the Atari disk drive. This disk should only have the DOS.SYS and DUP.SYS files on it. (The Atari Master Diskette is fine).

5. Using the DOS "Duplicate Disk" option (J), duplicate disk 8 to disk 1. This will copy the entire Atari disk to volume #1, including the "boot sectors" we need on volume 1 to get things rolling at powerup time.

We now have a "boot volume" (a volume that the Atari can start up from) on volume #1. We need to copy all the Corvus files from Volume 2 back to Volume 1, IF you want them back on volume 1. There is no particular reason (contrary to the Corvus manual) why they should have to be there, and given how long some of the Corvus programs are, it might be a good idea to file them elsewhere.

Now, let's try starting the system up from the Corvus without using the Atari 810. Turn the Atari off, and install the Integrater board (see Installation). Remove any cartridges. Next, turn the Atari on. In an extremely short amount of time the Atari DOS menu should appear; the Atari should boot directly from the Corvus. (You will be able to see the Corvus "Busy" light flickering as information is read off the disk.)

Congratulations! You have just booted the Atari from information on the Corvus. If you do a directory, you will find that the DOS.SYS and DUP.SYS files that you copied to volume #1 are indeed on drive #1.

From here, your use of the Corvus is application dependent. Here are several notes on the usage of the Integrater board.

The Corvus "Mount" program gives you the option of setting up a default mount table, of which Corvus volumes and Atari drives are attached to what Dn: window number. The Integrater uses this information when the Corvus first boots. Hence, if you specify Corvus volume #20 (for instance) as D1:, the system will boot off of volum #20. This is very important; this is how you get the Corvus to boot off of any volume.

You should install whatever DOS you feel comfortable using in the "boot volume" for the system to start up from. This is not something you are stuck with, however; if you hold down the OPTION key during poweron, you will be taken to the Mount Table Editor. There, you can respecify the "boot drive" (D1:) to make the system boot from any volume. Remember, what "Mount" says is the mount table is only used as a default mount table; you can override that any time with the Mount Table Editor.

When you need to "boot" a non-Atari DOS application (for example, a game or FORTH), sector copy the application to the volume you wish to boot from. Then, at powerup time, select that volume as D1:. The system will boot from that volume with no problem, the same way it would boot from an Atari 810. Note that with many applications, a standard DOS "Duplicate Disk" will not copy an entire nonstandard format to the Corvus; a sector copier must be used. There are many of these on the market.

"Copy protected" disks probably cannot be copied to the Corvus, as certain sectors of these disks are invalid. It is possible that if you contact the company producing the software, they will release you an unprotected copy for use on your Corvus.

Contact us if you have a special application which requires copy protected software to be placed onto the Corvus and perhaps we can work something out with the company involved.

Using the Corvus with copy protected software means using the Corvus as a database storage area. If the software allows more than one drive to be used, there is no problem; boot up from the copy-protected software disk (set D1: = Atari drive #1) and set the other Dn:'s to Corvus volumes. If it only allows one drive, you need to do something like this:

1. Put your copy protected disk into Atari drive 1.

2. Hold down OPTION & power on the Atari.

3. Select "1A1"; i.e., attach Atari drive 1 to D1:. This will allow the copy protected software to boot up.

4. When the software has finished booting, and is ready to use a data disk, do some sort of disk access (typical: search for an item on the disk) with OPTION held down. Attach whichever Corvus volume you desire as drive #1, replacing the Atari drive as D1:.

Remember that we have gone to great lengths to make the Mount Table Editor "invisible" to whatever routine is in memory. It restores all memory locations used in the Editor to their previous values, freezes the system clock (RTCLOK), unhooks the vertical blank vectors, and so on.

As we said before, you will have to experiment with the Editor and with the Corvus for your particular application; Corvus applications vary widely, and we can't possibly cover them here. However, you should now have the proper tools to use the Corvus as it was meant to be used, as an integrated part of the Atari operating system.

Mount Table Editor
for RAMDISK & Corvus

During any disk access, you can press the "OPTION" key and get to the Mount Table Editor, part of smallDOS. This Editor is used to change (edit) which window # a device is assigned to.

When you power up, the RAMDISK (if present) is assigned to window #2 and Atari drives 3-8 are assigned to windows 3-8. Go ahead and start the system up, holding down the OPTION key. (Remember you can press the OPTION key during any disk access and get this same effect; powering up the system forces a disk access, that's all. So does doing a directory, saving/loading a file, and so forth. We chose doing a disk access with OPTION pressed as the most generic way of triggering the Mount Table Editor to come in).

If you have a Corvus connected, the default Mount Table will be whatever you have specified with the Corvus program "Mount".

When the Mount Table Editor starts, it saves in some detail the state of the machine. The Atari can be considered "frozen" while the Mount Table Editor is active.

So, turn on the machine. A green display with all sorts of things on it shows up. Let's take them one at a time.

First, there's the "Mount Table Editor" and copyright notice from DSSA. Next, there are instructions for use. Next, there's a display of what window is hooked to what device.

Look over the display. D1: is attached to (look underneath)," A 1", which means Atari physical disk drive #1. So are all the rest of the Dn's:, except D2:. It is attached to "RAM", which means RAMDISK. If you have a Corvus attached, the display may say "C" and then a volume number, typically 1 ("C 1"), which means that D1: is attached to Corvus volume #1. The default Corvus arrangement is to attach D1:-D7: to Corvus volumes 1-7 and to attach D8: to Atari drive #1.

Assuming you have a RAMDISK, let's say we want to make the RAMDISK attached to D4:, not D2:. (If you don't have a RAMDISK, please read through this part anyway; it is much the same for the Corvus). First, let's reassign D2: to Atari drive #2.

Following the instructions onscreen (so you won't need to look through this manual every time you wanted to do this, we provided complete instructions onscreen), First, press a "2" (don't press RETURN! No need..). You will see the space where D2: is assigned to lights up to solid white, indicating you have the machine's attention. We have just specified which window we want to alter. Next, again following the instructions, type "A", to assign an Atari drive there. You'll see the "A" show up in the window. Finally, type a "2", and the lower display will now show "A 2" right below "D2:". Now, when you make an access to D2:, it will go to Atari physical disk drive #2.

Now, let's assign the RAMDISK to drive 4. Type "4" to let the Atari know which window we want to edit, and "R" to let it know we want the RAMDISK attached there. It will plot "RAM" under D4: to let you know that window #4 is now attached to the RAMDISK. Incidentally, no data in the RAMDISK will be destroyed by this move; for instance, if you copy data to the RAMDISK when it is D2:,. that data will still be there when the RAMDISK becomes D4:.

Assuming you have a Corvus, let's switch volume numbers. Let's assume that things are in their default state, where D2: is connected to Corvus volume 2 and D4: is connected to volume 4. Let's swap them, so D2: will talk to volume 4 and D4: will talk to Volume 2. (Note that volume 50 or whichever could just as easily be used).

First, type 2C4 (return). This attaches Window #2 (D2:) to Corvus volume 4. Then, type 4C2 (return). This attached window #4 (D4:) to Corvus volume #2. From then on, any disk access to D2: or D4: will go to the new destination.

The Corvus volume # can be any from 1 to the maximum volume number on your Corvus drive, which varies according to the drive size. The maximum volume number is given to you at the bottom of the screen, for your reference. (Note: if this line ever fails to show up, it means the system does not know the Corvus is on line. For RAMDISK users without Corvus's, this line will thus never appear).

If you want to assign an Atari drive to a given window number, use "A", then drive number. For instance, to assign Atari drive #4 to Window #3, use 3A4 (return). The display will show the "A 4" (for Atari drive #4) under window #3.

It is confusing at first to keep the ideas of the window versus a physical device straight, but as soon as you feel comfortable with the idea, the power of the Mount Table Editor will be greatly increased for you.

How to get out of the Mount Table Editor? Type "X" (no return needed), and you'll bounce back into your program whereever you were when you exited. Nothing should be disturbed; great pains have been taken to ensure this is true.

You can also have the RAMDISK assigned to several different windows, with the same data appearing at each window. There will not be seperate RAMDISKs at each window, but the same RAMDISKs data on each Dn:. You can also assign the same Corvus volume to different windows with essentially the same results.

For example, boot up with a DOS configured for 4 drives.

(How? Very well. Boot up with a regular DOS 2.0S disk and BASIC. Type POKE 1802,15. Press RETURN, then RESET. Go to the DOS menu, format a disk, and write DOS files to it. This new disk will be configured for 4 drives. This POKE 1802 business is just letting the DOS know how many windows to open; in this case, 4. If ytou ever get an ERROR-160, you are trying to use a window that the DOS hasn't been set up for. See the Appendix in the Disk Operating System Manual for more information)..

Go to DOS. Do a directory of drive 1, holding down the OPTION key. You'll pop into the Mount Table Editor. Type 3R to assign window #3 to the RAMDISK and 4R to do the same with D4L:. Next, type "X" to exit the Mount Table Editor. You'll be back at the Atari DOS menu. Type J to Duplicate Disk from 1 to 2 ("1,2"), and duplicate the drive to the RAMDISK. (You'll be interested to see how long it takes to read the data in from the Atari disk versus how long it takes to write it out from the RAMDISK.) Next, do a directory of D2:; it should match D1:, your Atari diskette. Do a directory of D3:, which we have also made the RAMDISK, and it too will be the same; in fact, if you do a J (Dup Disk) from 2 to 3, you will duplicate the RAMDISK on top of itself. This will happen very quickly.

This is just a short example of the power of Mount Table Editor. Remember, from whereever you are in any software, if you cause a disk access and hold down the OPTION key, you'll be taken to the Editor. And once again, instructions on its use are onscreen; just follow the steps outlined there. We believe that once you try the examples here and experiment some on your own, you'll not need this manual again. (Personally, I believe that a manual that only needs to be used once is an ultimate manual).

Incidentally, if you ever boot up without the RAMDISK, you will note that the Editor won't have it as D2:'s assigned device. Rather, regular old Atari Drive #2 will be assigned there. And if you try to assign the RAMDISK anywhere when it is not plugged in, a "beep" will be sent to let you know it can't be done.

Similarly, if you try to assign a Corvus volume without a Corvus being attached, you will get the warning beep.

Any illegal keyboard press will result in a "beep", by the way. If you get one, back up and look at what you've typed, and try again.

If the Mount Table gets screwed up (typically, by cycling the power off and on very quickly, so that glitches get into the tables) the message ERR may show up. This means it is time to straighten out the Mount Table, either with the editor or elsewhere. This can also occur if you assign things illegally using the direct editor calls from BASIC listed below. As an example, you could assign Atari drive #0 to a given window; this will generate an illegal message.

Since smallDOS searches for a connected Corvus at powerup, there may be a one second delay on powerup before much begins to happen; this is normal. (The Atari is waiting for a Corvus to send it a Ready signal, and not receiving the signal; after a second, it gives up and decides there is no Corvus attached). Note that if you powerup the Corvus and the Atari at the same time, the Corvus waits around 30 seconds to become "alive"; during this time, the Atari will not find it online. You will need to wait until the Corvus is ready before powering on the Atari.

Please feel free to experiment with the Mount Table Editor and report and bugs you find to us; we know of none presently, but possibly there are some.

In summary, we feel you will find that having the RAMDISK's power on tap at any Dn: at any time, and having that assignment be changed at any disk access will greatly increase the usefulness of your RAMDISK and Corvus.

Automatic Mount Table Switching

The Mount Table Editor is certainly a convenient way for swapping around disk allocations. However, there are a great many places where automatic switching might be desirable. If this is the case, the following built in routine will be a great convenience for you.

If you should need to swap window assignments from within a program, do it like this:

```
X=USR(49152,window #,15 for RAMDISK or Atari Drive #,15)
```

or for Corvus,

```
X=USR(49152,window #,128, volume #)
```

The 128 tells the routine to attach a Corvus volume to the window; any number less than 128 will be interpreted as an Atari drive, so be sure you give it legal parameters. If you don't, nothing terrible will happen, except that window just won't work.

To make the RAMDISK assigned to window 3:

```
X=USR(49152,3,15,15)
```

To assign Corvus volume 60 to window 4:

```
X=USR(49152,4,128,60)
```

To make Atari drive #4 attach to window #1:

```
X=USR(49152,1,4,15)
```

From assembler, call $C000 with the standard BASIC stack protocol for the above parameters, including the initial # of arguments. Note this means two values pushed on the stack for each parameter.

Custom Programming

If you should requie some custom programming for either the RAMDISK or Corvus, feel free to contact us. We do all sorts of custom work for a number of firms around the country. You may find that you do not have the time to learn about the Corvus in sufficient detail to (for instance) set up a multi-volume database; we have done this sort of thing many times, and can provide you with a speedy solution.

School systems in particular may wish a "custom" Integrater board. Students can be locked out of certain areas of the Corvus, and those areas can then be used for student record keeping with very high security. Then, a password protected Integrater can be used as the instructor's terminal to access these records. We can also set up an automatic name-based mount table where a student enters their name and is then assigned to a partitioned area of the Corvus, where their work cannot be accidentally or deliberately altered by another student.

Custom Integrater boards can also perform other functions. For instance, a "front ports" printer cable can be constructed which is completely compatible with today's software and does not require the Atari 850 Interface, a device that is becoming increasingly difficult to obtain.

Again, feel free to contact us to discuss your special needs. We realize that while the Integrater is powerful and very generic, there are applications where some customizing can be of great value.

Warranty

We warrant the Integrater to be free from defects in parts and workmanship for 90 days from the date of purchase. We cannot be responsible for user modifications to the Integrater as shipped and such modifications void this warranty. We will repair Integraters in warranty at no charge within this 90 days and pay shipping one way. This warranty is intended for parts that fail during the "burnin" period or possible intermittent or defective boards; all Integrater boards are extensively tested before shipping.

After 90 days, we charge by the hour and the customer is expected to follow standard depot maintenance procedure; i.e., ship both ways. We will repair or replace your Integrater board as quickly as possible. Please call before returning your board both for diagnostics we can run over the phone and for a return authorization number; we cannot accept boards without this number.

Ron -

The Integrater is produced by:

ADS
9202 Cedar Crest Drive
Austin, Texas 78750
512-258-6676

Software & Hardware design by D.S.S.A.
(David & Sandy Small and Associates)

Sales Support by Adams Computer Systems, INC.

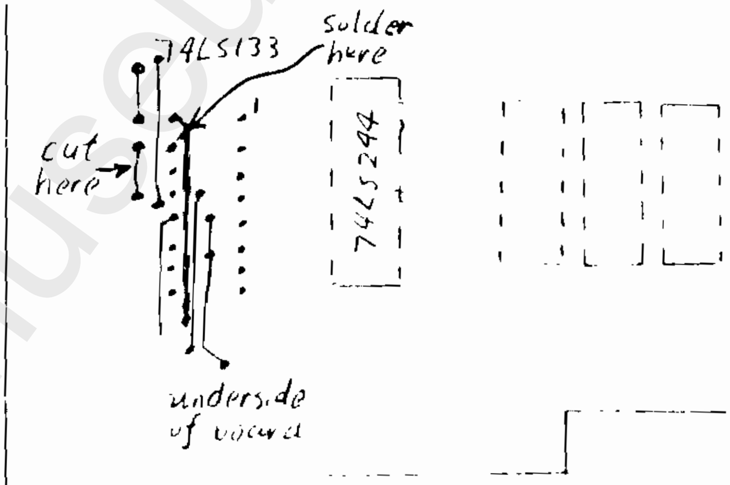
LS138
chip most likely
to go bad

RAMDISK mod

Another very useful modification can be made to your RAMDISK board to make it much better behaved. It will save you from having to flip the switch on the RAMDISK in order to run software which loads into the $FC0-$FFF range. As it stands, writing to this address range will cause RAMDISK banks to be switched in in place of user memory. This usually causes the program to crash. This mod disables the $FC0-$FFF range and doubles the other select range from $CFC0-$CFFF to $CF80-$CFFF, which should cause no problem. However, since this mod may void your warranty you may want to wait until it expires.

1) Locate pin 18 on the card edge of the 16K RAM board in slot 1. This is the select line for the lowest 8K and should go to Z501 pin 1. Solder a 6" wire to the indicated pin of this IC.

2) Locate the 74LS133 IC on the RAMDISK. Solder the other end of the wire to pin 15 of this IC. Make the indicated cut on the underside of the RAMDISK board and the mod is complete.



Ramdisk Mod