

SCRAM 2000

User's Manual

Note - when you buy the PCB, you get a nice bound manual with diagrams and pagenumbers even! This file is the textual contents of the manual.

CONTENTS

Introduction	3
Features	4
Installation	5
Installing a Hard Card	6
Installing an Internal SCSI Device	7
Connecting an External SCSI Device	7
SCSI Address ID	7
SCSI Terminators	8
Terminator Power	9
External SCSI Connectors	10
AutoBoot	11
AutoBoot Disable	11
SCRAM 8/16 Bit Controller Option	12
RAM	
Installing RAM	14
Setting the RAM jumpers	15
Using 4 Meg ZIPs	16
Diagnostic LED Operation	17
Custom Logic	18
Sample Driver Code	20
SCRAMPREP Installation Utility	23

Introduction

Congratulations on choosing the **SCRAM 2000** SCSI/RAM controller for your Amiga 2000. This unit has been designed to give you maximum performance from your Amiga 2000 with the minimum of fuss.

Please take some time to read the information in this manual, as it contains important tips which will guide you through the installation procedure. This is especially important for users who are installing their own DRAM chips or hard drives. Check your setup against the manual to get it right first time!

Setting up a SCSI drive has never been easier than with **SCRAMPREP** installation software. This utility gives you an unprecedented level of control over disk partitioning and **RigidDiskBlocks** while guiding the novice through on auto-pilot.

As you know, the **SCRAM 2000** controller is a dual SCSI controller which supports either an 8 bit SCSI chip, or a super-fast 16 bit device. If you have the eight bit version you can always upgrade to the faster 16 bit system with the **SCRAM 2000 SuperCharger Kit**.

Finally, if you are new to the Amiga or this is your first hard disk, please read the **AmigaDOS** manual to familiarize yourself with DOS procedures.

Features

The **SCRAM 2000** controller card has the following advanced features.

RAM

- o 8Mb FastRAM controller configured as 0Mb, 2Mb, 4Mb or 8Mb
- o Low cost ZIP DRAMs - 256Kx4 or 1024Kx4 - jumper selectable
- o DRAM is socketed for easy installation by user
- o Full speed 0 wait-state FAST RAM
- o Jumper select RAM size for AUTOCONFIGURE including disable
- o RAM Access LED
- o AutoConfigure LED

SCSI

- o AutoBoot ROM driver supports Rigid Disk Block
- o CD-ROM compatible with ISO 9660 File System
- o SYQUEST removeable media with advanced DiskChange
- o Toggle switch to disable AutoBoot function
- o External DB25 SCSI connector
- o SCSI Access LED
- o Power option for external SCSI termination
- o Non-DMA transfer technique for complete system compatibility
- o High Speed data transfers with custom SynchroScsi logic

16 BIT SCSI

- o Optional Hi-Performance 53C94 SCSI controller
- o SCSI-2 compatible ANSI X3.131 -1986
- o 16 byte FIFO and full word transfers
- o Asynchronous transfers up to 4MB/S, Synchronous to 5MB/S

SOFTWARE TOOLS

- o New **SCRAMPREP** utility gives total control over SCSI devices
- o Supports Rigid Disk Block format for industry standard devices
- o Low level format
- o Bad Block mapping
- o SCSI Snooper thoroughly analyses all attached SCSI devices

Installation

Carefully unpack the **SCRAM 2000** shipping carton. You should find the following items:-

- o **SCRAM 2000** controller for the Amiga 2000
- o **SCRAMPREP** Installation diskette
- o Mounting hardware for Hardcard option
- o **SCRAM 2000** User Manual

After turning mains power off and disconnecting the power cord, remove the cover of your Amiga 2000 and prepare a static-free surface to place the **SCRAM 2000** card on. A conductive Anti-Static mat with wrist strap is ideal, however if this is not practical then at least make sure you touch the machine chassis to ground yourself before handling the SCRAM card or any electronic components. Be wary of working on carpet - timber or concrete floors are better. Some synthetic clothing and footwear may also cause static buildup. If the air is dry and you are getting zapped then you are likely to damage your computer.

Before the **SCRAM** unit is plugged into your computer, you may have to install RAM. In this case it will be necessary to configure the various options on the **SCRAM 2000**. Check the relevant sections of this manual if you are unsure. If your **SCRAM 2000** has been shipped with RAM installed, then there should be no need for you to modify the jumper settings. The unit will have been fully tested at the factory.

Make sure the Jumpers (3 of them) associated with RAM are set correctly. If the board is optioned incorrectly the system may hang on power on. If you are installing your own DRAM chips then consult the section - Installing RAM.

If you are installing the **SCRAM 2000** as a Hard Card, then you will have to mechanically mount a 3.5" SCSI drive on the board and attach power and data cables (supplied). Refer to the section - **Installing a Hard Card**.

Plug the SCRAM card into a spare 100 pin slot and screw down the end plate. The first slot is probably the best if you are installing a full height drive Hard Card to avoid wasting a slot due to clearance problems.

Don't forget to connect the external HD LED to the **SCRAM 2000** card

- located just above the RAM jumpers.

Installing a Hard Card

The **SCRAM 2000** has mounting space for a single 3.5" SCSI hard disk drive such as the Quantum ProDrive LPS series. The drive is mounted at the rear of the card with two short cables - 4 Pin Power and the 50 Way SCSI bus. When mounting the drive take care to use the threaded spacers provided to raise the drive above the printed circuit board. This will avoid the possibility of shorts and allow some air circulation to prevent overheating of the drive electronics.

When mounting the **SCRAM 2000** HardCard in the Amiga it will usually be best to locate it in the first slot (closest to the Power Supply). This position provides maximum clearance for the Hard Drive, and in the case of a full height unit (more than 200 Megs) will not waste one of the Amiga slots.

Make sure you firmly secure the metal end-plate of the SCRAM card to the back of the Amiga 2000 when a hard drive is mounted. This will prevent the weight of the drive from dislodging the card edge connector seated in the 100 pin socket on the motherboard.

Installing an Internal SCSI Device

The **SCRAM 2000** is capable of controlling several internal SCSI devices in addition to external devices and a Hard Card. Normally there will be space for one 3.5" device and one 5.25" device in an Amiga 2000. Typical internal drives could be a 5.25" Syquest Removeable Media unit or a 3.5" Floptical Drive.

SCSI connector JP4 (the 50 way header along the top of the board) is provided for connection to internal SCSI devices. In the case of two internal devices, the SCSI cable will have to have an extra header at an appropriate position.

The jumper JP2 will optionally supply +5 Volts (current limited through R4) to pin 26 of the SCSI-INT connector. This option is not normally required.

Connecting an External SCSI Device

External SCSI devices such as the SCRAM Drive may be connected via the Industry standard DB25 connector on the end plate of the SCRAM unit. Make sure you consider the location of SCSI terminators and set unique SCSI addresses for all external devices. This is the most likely cause of difficulty in setting up multi target SCSI systems.

Jumper JP3 optionally provides +5 Volts (current limited through R3) on pin 25 of the DB25 connector.

SCSI Address ID

Up to seven SCSI devices (Targets) may be connected to the **SCRAM 2000** (Host). The SCSI bus uses an address of 0-7 to identify different Targets. Typically addresses 0-6 are used for Targets and 7 is reserved for the Host.

SCSI devices such as Disk Drives will normally have a set of three jumpers to set the SCSI address - although you should note that Hard Drives will normally be shipped set as SCSI ID 0. When configuring your system you must ensure that every SCSI device is given a unique ID - there must be no SCSI address conflicts for correct operation.

The **SCRAM 2000** AutoBoot driver software will interrogate all SCSI addresses on the Bus at boot time and will boot from the first bootable partition. When the drive(s) are prepared with **SCRAMPREP** each drive will have its RDB written on the first tracks, avoiding

the need for MountList entries. The RDB convention is now common across all Amiga SCSI controllers so you may use a drive prepared on another controller with the **SCRAM 2000** without reformatting.

SCSI Terminators

The SCSI Bus is electrically specified to have Termination resistors at both ends of a 50 ribbon cable. The **SCRAM 2000** is normally considered to be one end (Host) of the SCSI Bus and has sockets for three 220/330 SIP parts. These SIPs will normally be installed on the **SCRAM 2000**. Other SCSI devices such as Hard Drives will also have terminators installed as shipped, and these may be removed or disabled. You will need to refer to the installation data to locate the terminators for the particular drive.

The SCSI specification allows only two sets of terminators - one at each end of the bus. In practice the SCSI bus may not be a simple length of cable and you will need to experiment to determine the best location for the second Terminator in the case of multiple SCSI units.

Examples:-

- o 1 Hard Card SCSI drive leave terminators on both SCRAM 2000 and SCSI drive unit
- o 1 Hard Card SCSI drive leave terminators on internal 1 Internal SCSI device drive and either the **SCRAM 2000** or the Hard Card drive.
- o 1 Internal SCSI drive leave terminators on **SCRAM 2000** and SCSI device.
- o External SCSI drive terminate the External drive and only one of either **SCRAM 2000**, Hard Card or Internal drive.

Terminator Power

The **SCRAM 2000** card has jumper options to supply +5 Volts (current limited) to the external SCSI connectors. This allows the SCSI bus to be terminated with a passive Terminator Plug which may be supplied with some SCSI peripherals.

External SCSI Devices

The **SCRAM 2000** card has an External SCSI connector (see Figure 3) on the rear panel. This connector is a DB25 Socket to accept the standard 25 -> 50 way SCSI cable. Usually external SCSI devices will come with a suitable cable, however most computer stores or mail order shops can supply them.

AutoBoot

The **SCRAM 2000** controller is designed to support the Amiga AutoBoot facility which now includes the Rigid Disk Block convention. The **SCRAM 2000** driver software is supplied as firmware on the EPROM (U32). This EPROM is recognised at AutoConfig time and the system will load the driver into RAM and continue with the Boot sequence. The Amiga will look for bootable devices such as DF0:, Hard drives and Networks and will boot from the highest priority one. If no floppy disk is present in DF0: then the Amiga will typically boot from the Hard Drive.

AutoBoot is only supported from Kickstart 1.3 on. If you have an Amiga 2000 with Kickstart 1.2 in ROM you will have to either upgrade to 1.3 or disable AutoBoot and use the driver supplied on the **SCRAMPREP** disk. See the Kickstart 1.2 section for details of using the floppy based driver.

AutoBoot Disable

The **SCRAM 2000** features a toggle switch at the rear of the machine to disable the AutoBoot facility. This switch is provided for software compatibility with Kickstart 1.2 ROMs or for special cases such as games where it is necessary to boot from floppy. It must be remembered that with AutoBoot enabled, the driver software in EPROM will be loaded into RAM regardless of where the system boots from. This may cause problems with Game software which needs all of the computer's memory.

AutoBoot is ENABLED when the toggle is to the LEFT. AutoBoot is DISABLED when the toggle is to the RIGHT.

NOTE: If you have Kickstart 1.2, the AutoBoot must be off or the machine will not boot correctly!

SCRAM 8/16 Bit Controller Option

The **SCRAM 2000** may be configured as either a standard 8 bit SCSI interface, or as a high performance 16 bit SCSI adapter. In Fig. 6 you can see that there are two sockets for plugging in SCSI controller chips. The 44 pin socket at the top of the board will accept an 8490 (enhanced 53C80) chip. The larger 84 pin socket directly below it is for an NCR 53C94 16 bit SCSI chip.

The 53C94 internally converts the 8 bit data stream from the SCSI device to a stream of 16 bit words for the Amiga's 68000 CPU. This more efficient word transfer in addition to a

16 byte FIFO buffer in the 53C94 means much higher performance data transfers between the Amiga 2000 and SCSI devices.

In a normal 68000 Amiga, the 16 bit **SCRAM 2000** will perform about 50% ahead of an 8 bit **SCRAM 2000**. In an accelerated Amiga environment (68030) the 16 bit SCRAM can support sustained data transfer rates of up to 2Mb/sec.

The difference between a SCRAM-8 and a SCRAM-16 is merely which controller chip and associated support chips are plugged in. The SCSI chip in place must have the correct Boris, Cyril and SCRAMBoot ROM. There are no jumper options to differentiate between an 8 bit or 16 bit board - its all in the chips.

RAM

Installing RAM

The **SCRAM 2000** may be configured as a RAM card using Industry standard 256Kx4 or 1Megx4 ZIP DRAMs. All the RAM chips are socketed so the user may purchase chips to upgrade the RAM size.

Make sure you plug the ZIP RAM chips in the correct orientation. As you look at the board with the edge connector at the bottom, the RAMs will plug in with PIN 1 to the left (edge of the board).

WARNING! OBSERVE STATIC PRECAUTIONS WHEN HANDLING SCRAM 2000 AND RAM DEVICES

Setting the RAM Jumpers

Set the RAM jumpers according to Fig. 7 for correct RAM configuration. The setting of these jumpers is irrespective of the type of DRAM used, however 0M or 2M are the only valid settings if using 256Kx4 ZIPs. These jumpers should be orientated vertically (for setting RAM size), whereas the 4Meg jumper is horizontal. The correct jumper settings are also indicated on the PCB.

Make sure you always populate RAM from BANK 0 through to BANK 3, in that order. This means the first 4 ZIPs you install go in sockets marked "0", the next 4 in sockets "1" and so on.

RAM options:-

SIZE	CHIPS	RAM TYPE
2Meg	16	256Kx4 ZIP DRAMs
2Meg	4	1Megx4 ZIP DRAMs
4Meg	8	1Megx4 ZIP DRAMs
8Meg	16	1Megx4 ZIP DRAMs

The table above shows the various ram options available with the **SCRAM 2000**. The RAM controller may be totally disabled by selecting the 0Meg jumper option.

Using 4 Meg ZIPs

If you are installing 1Megx4 DRAMs you must short the 4 Meg jumper. This changes the RAM logic to accomodate the extra address lines associated with these chips. If you short this jumper with 256Kx4 DRAMs plugged in, you will encounter RAM malfunctions.

The 4 Meg jumper is located just above the RAM Size jumpers at the top left of the **SCRAM 2000** board - refer Fig. 6.

When populating the RAM sockets, always fill banks 0 to 3 in that order. The RAM Bank number is clearly printed on the PCB.

You must not mix the 256Kx4 and 1Megx4 chips - if you are using the 4Meg chips short the jumper "4M Chips"! Set the RAM jumper block for the amount of memory you are installing. The table of options is printed on the board just below the jumper pins.

Typical Devices:

MICRON	MT4C4001Z-80
TOSHIBA	TC514400Z-80
FUJITSU	MB814400Z-80
NEC	uPD424400V-80
OKI	MSM514400Z-80
HITACHI	HM514400Z-80

If you are sourcing your own RAM chips make sure you get the right type. Specify ZIP package, 1024K x 4 and Page Mode. Any speed device should work, however 80nS or 100nS devices will be the cheapest.

Diagnostic LED operation

The **SCRAM 2000** has three LED Indicators visible on the top of the board. These LEDs will flash during operation of the unit and provide information about the system for diagnostic purposes when you are installing the board or troubleshooting. Only the DISK LED is visible externally and this is connected to the Amiga 2000 front panel through jumper "HD".

RAM

Red LED is active when the Amiga is accessing FAST RAM on the **SCRAM 2000**. This dynamically indicates System activity and can be a very useful guide to the operation of your software.

DISK

Orange LED is active during the data transfer phase of SCSI transactions. This LED can be used as the Disk activity light in cases where the SCSI drive is not easily seen. This LED is the same as the front panel DISK light.

ACTIVE

Green LED is turned on at the end of a successful AutoConfigure operation. When the Amiga is powered on or rebooted, this LED will be off. If the system is working normally, this LED will come on within about 1 second. Any system malfunction or reset will extinguish this LED.

Custom Logic

The **SCRAM 2000** SCSI RAM controller achieves a high level of integration by using five custom logic chips. These chips implement most of the circuit operation, thereby providing a low component count and subsequent low cost. In the event of a circuit fault, most of the board logic is socketed ensuring repairability. The overall function of the custom chips is described below.

Boris 8/16 controls the AutoConfigure logic of the **SCRAM 2000**. This chip describes the RAM and IO sections to the Amiga, sets RAM size and base offsets.

Cyril 8/16 is the SCSI AutoBoot controller chip. This chip manages data transfers to the 8490V or 53C94 SCSI controller, block transfers, interrupts, bus synchronization and Disk LED.

Griswold is the master timing controller for the DRAMs. This chip generates RAS and CAS, refreshes the RAMs and synchronises with the 68000.

Ronald controls RAM operation including address decoding, bank selection, 1M/4M selection and XRDY operation.

Dennis is the main address decoder for the **SCRAM 2000**. From the base address and the RAM size, this chip will locate the banks of RAM in physical address space.

Sample Driver Code

```
*=====*
```

```
*
* Example program demonstrating the various methods of talking to
* the SCRAM 2000 device driver.
*
* Both standard and SCSI-Direct methods are demonstrated.
*
* The code is verbosely commented to make it simple to understand
* but some knowledge of 68000 assembler programming on the part of
* the reader is assumed.
*
* All code by Will McGovern. (yes, I wrote the driver as well !!)
*
* BLATENT AD ==> For a custom written driver for your hardware,
* send inquiries to :
*
* Will McGovern
* PO Box 247,
* NEW LABTON,
* NSW 2289
* AUSTRALIA
*
* The code is written to be assembled with Macro68 from DigiSoft
* but can be converted to other formats with ease. Any Macro68
* specific directives are explained.
*
* The 2.0 include files are used for all symbols and equates.
*
* THIS CODE IS COMPLETELY PUBLIC DOMAIN. USE IT HOWEVER YOU WANT !!
*=====*
```

Macro68 assembler directives

```
mc68000 ;68000 mode
strict ;strict syntax mode
exeobj ;executable object file
objfile 'example' ;object filename
```

SYS : Call system vector macro

```
sys macro
jsr(_LVO\1,a6)
endm
```

```
*-----*
```

```
EXAMPLE_UNIT equ 0 ;scsi unit to talk to
BLOCK_SIZE equ $200 ;size of 512 byte block
MAXAUTO_SIZE equ $fe ;maximum autosense size
MAXINQUIRY_SIZE equ $fe ;maximum # of inquiry bytes
```

```
*-----*
```

```
section excode,code
```

Find this task and see if we started from workbench or from a CLI

```

start      movea.l   (4).w,a6           ;exec library base
          suba.l   a1,a1           ;this task
          sys     FindTask        ;find this task
          movea.l  d0,a4           ;save task pointer
          tst.l   (pr_CLI,a4)     ;did we come from a CLI ?
          bne.s   clistartup      ;branch if CLI entry

```

Discard the workbench startup msg

```

          lea     (pr_MsgPort,a4),a0 ;this task's message port
          sys     WaitPort          ;wait for WB startup message
          lea     (pr_MsgPort,a4),a0 ;this task's message port
          sys     GetMsg           ;fetch the startup message

```

Initialise a message port (MP) for use with my IORequest structure

```

clistartup movea.l   #mymp,a2           ;my message port
          move.l   a4,(MP_SIGTASK,a2)   ;save pointer to this task
          moveq    #-1,d0              ;any signal will do
          sys     AllocSignal          ;allocate a signal for MP
          move.b   d0,(MP_SIGBIT,a2)    ;save signal # in MP
          bmi     nosignal             ;branch if error
          movea.l  a2,a1              ;copy MP pointer
          sys     AddPort             ;add my MP to the system

```

Open scram.device for EXAMPLEUNIT

```

          movea.l  #scramname,a0       ;scram.device name
          movea.l  #myior,a1          ;my IORequest structure
          move.l   a2,(MN_REPLYPORT,a1) ;init MP pointer in IOR
          moveq    #EXAMPLE_UNIT,d0    ;scsi unit to talk to
          moveq    #0,d1              ;no flags
          sys     OpenDevice          ;open scram.device
          tst.l   d0                  ;any errors ?
          bne.b   noscramdevice       ;branch if error

```

* Now we can talk to the scram.device through the standard
* device commands such as CMD_READ or use the HD_SCASICMD command
* for SCSI-Direct mode.

Here are some examples of normal and SCSI-Direct mode access.

Read block 0 from the unit into blockbuffer using CMD_READ command

```

          movea.l  #myior,a1           ;IORequest pointer
          move.w   #CMD_READ,(IO_COMMAND,a1) ;CMD_READ command
          move.l   #blockbuffer,(IO_DATA,a1) ;buffer for data
          clr.l   (IO_OFFSET,a1)      ;block 0
          move.l   #BLOCK_SIZE,(IO_LENGTH,a1) ;one block to read
          sys     DoIO                ;read the block
          tst.b   d0                  ;any error ?
          bne.b   cmdreaderror        ;error if d0 not zero

```

Now do the same as above in SCSI-Direct mode

```

          movea.l  #myior,a1           ;IORequest pointer
          move.w   #HD_SCASICMD,(IO_COMMAND,a1) ;CMD_READ command
          move.l   #scsireadcmd,(IO_DATA,a1) ;pointer to SCASICmd
          sys     DoIO                ;read the block
          tst.b   d0                  ;any error ?

```

```
bne.b    scsireaderror                ;error if d0 not zero
```

Perform a SCSI INQUIRY command on the EXAMPLE_UNIT

```
movea.l  #myior,a1                    ;IORequest pointer
move.w   #HD_SCsICMD,(IO_COMMAND,a1)  ;HD_SCsICMD command
move.l   #scsiinquirycmd,(IO_DATA,a1) ;pointer to SCsICmd
sys      DoIO                          ;perform inquiry
tst.b    d0                            ;any error ?
beq.b    exitexample                  ;error if d0 not zero
```

This is where an error handler would be placed if this was serious code

```
cmdreaderror:
scsireaderror  nop
```

Clean up our mess and return to DOS

```
exitexample    movea.l  #myior,a1        ;pointer to my IORequest
               sys      CloseDevice    ;close scram.device
noscrampdevice movea.l  #mymp,a1        ;pointer to my MP
               sys      RemPort        ;remove my message port
               moveq    #0,d0          ;prepare D0 for byte load
               move.b   (mymp+MP_SIGBIT),d0 ;get signal # we allocated
               sys      FreeSignal     ;free the allocated signal
nosignal       moveq    #0,d0          ;clear return code
               rts                    ;return to DOS
```

*-----

```
section exdata,data
```

Note: The SCsICmd structure used in this example did not appear in the early 1.3 include files in its entirety. C= omitted the autosense information. See the 2.0 include file "devices/scsidisk.i" for a full description of the SCSI-Direct protocol.

Also note that the SCsIF_AUTOSENSE (4 byte sense length) has become SCsIF_OLDAUTOSENSE in the 2.0 implementation.

The new SCsIF_AUTOSENSE supports sense data lengths of 4 to 255 bytes. The sense length is specified in scsi_SenseLength field.

The SCsIF_READ/SCsIF_WRITE flags are NOT required by the scram.device as the data direction is determined automatically by the driver.

SCsICmd structure for reading block 0

```
scsireadcmd    dc.l      blockbuffer    ;data buffer address
               dc.l      BLOCK_SIZE    ;number of bytes to read
               dc.l      0              ;actual bytes read
               dc.l      readcmd        ;pointer to scsi CDB
               dc.w      10             ;# of command bytes
               dc.w      0              ;actual cmd bytes sent
               dc.b      SCsIF_AUTOSENSE ;automatic sense
               dc.b      0              ;status byte
               dc.l      sensebuffer    ;buffer for sense data
               dc.w      MAXAUTO_SIZE   ;size of my sense buffer
               dc.w      0              ;actual sense bytes read
```

SCSICmd structure for an INQUIRY command

```
scsiinquirycmd dc.l    inquirybuffer    ;data buffer address
                dc.l    MAXINQUIRY_SIZE  ;number of bytes to read
                dc.l    0                ;actual bytes read
                dc.l    inquirycmd       ;pointer to scsi CDB
                dc.w    6                ;# of command bytes
                dc.w    0                ;actual cmd bytes sent
                dc.b    SCSIF_AUTONSENSE ;automatic sense
                dc.b    0                ;status byte
                dc.l    sensebuffer      ;buffer for sense data
                dc.w    MAXAUTO_SIZE     ;size of my sense buffer
                dc.w    0                ;actual sense bytes read
```

```
* Here are the actual command descriptor blocks (CDB's) sent to the
* selected scsi unit.
*
* For more information on these consult the SCSI specifications or the
* manual for your scsi device.
```

```
readcmd        dc.w    $2800,$0000,$0000,$0000,$0100 ;extended read
inquirycmd     dc.w    $1200,$0000,MAXINQUIRY_SIZE<<8
```

Text and byte data

```
scramname      cstr    'scram.device'      ;nullterminated name
                even
```

*-----

```
section        exbss,data

mymport        ds.b    MP_SIZE              ;my message port structure
                even
myior          ds.b    IOSTD_SIZE           ;my IORequest structure
                even
inquirybuffer  ds.b    MAXINQUIRY_SIZE     ;inquiry data buffer
                even
sensebuffer    ds.b    MAXAUTO_SIZE        ;autosense data buffer
                even
blockbuffer    ds.b    BLOCK_SIZE          ;block data buffer
                even
```

*-----

end

SCRAMPREP Installation Utility

SCRAMPrep is a disk utility provided to work with your SCRAM 2000. You will need to use **SCRAMPrep** on a new SCSI Hard drive to partition it and install the Rigid Disk Blocks necessary for AutoBooting. **SCRAMPrep** also provides additional advanced functions for SCSI devices.

Quick Guide

1. Attach **SCRAM 2000** to the Amiga.
2. Connect Hard Drive to SCRAM with SCSI cable.
3. Switch AutoBoot ON (toggle towards computer).
4. Power on SCSI Drive and Amiga 2000.
5. Boot the **SCRAMPrep** Disk supplied with your unit.
6. Run **SCRAMPrep**.
7. Click select your drive from the table 0 - 6.
8. Select FULL AUTOMATIC PREP from control panel.
9. Select the number of partitions you require and write the data.
10. Quit **SCRAMPrep** and REBOOT with a WorkBench disk.
11. Format all the partitions using WorkBench Format!
12. Copy all of the WorkBench Disk to DH0:.
13. Remove WorkBench floppy and reboot from hard drive.

LOW LEVEL FORMAT

Selecting this option will cause **SCRAMPrep** to issue a Low Level Format command to the selected SCSI drive. This command is executed by the SCSI drive and will completely reformat the drive, erasing ALL data. Some drives (e.g. Quantums) ignore this command.

MAP BAD BLOCK

Selecting this option will cause **SCRAMPrep** to go out and verify every block on the SCSI Drive. If bad blocks are found, **SCRAMPrep** will tell the SCSI drive to map the blocks out. This option is a low level operation which can be very slow. Only do it if you don't want any data on the drive and are confident with what you are doing.

PARTITION DRIVE

This option gives you full manual control over partition setup. Using the Partition Panel which pops up, you can set the exact size of each partition.

INSTALL FILESYSTEM

This option allows you to write a file system of your choice onto a boot partition. Typically this will be AmigaDOS however, other FileSystems can be specified.

KickStart 1.2

If your machine has KickStart 1.2 ROMs in it, you cannot use AutoBoot with the **SCRAM 2000**. To get full use of your SCRAM you must upgrade to Kickstart 1.3 or later. It is still possible to use the **SCRAM 2000** with a 1.2 Amiga, however you will have to have a Boot floppy inserted in DF0:.

Detecting Kickstart 1.2

When you turn the machine on with no WorkBench floppy in DF0: you will see a hand requesting you insert a floppy. The Kickstart version number is written on this screen.

SCRAMPrep the Drive

If you are intending to use the **SCRAM 2000** with KickStart 1.2, then you will need to run **SCRAMPrep** first as for an AutoBoot system. This will partition the drive and set up the Rigid Disk Blocks. The drive will also be ready for AutoBoot when you upgrade your machine.

Prepare a Boot Floppy

Make a copy of your **SCRAMPrep** Disk to use as a Boot Disk. Put your master **SCRAMPrep** disk away.

On this new disk, rename Startup_Sequence.HD to Startup_Sequence.

This new Startup_Sequence will run a program called ScramAutoMount which will load a SCRAM driver from DEVS: and mount any partitions on SCSI devices connected to the SCRAM 2000.

Format your Partitions

The first time you boot with your new Boot Floppy, make sure you Amiga Format all hard disk partitions. This can be done from the Workbench or CLI - e.g.

```
system/format drive dh0: name fred FFS QUICK NOICONS
```

Notes