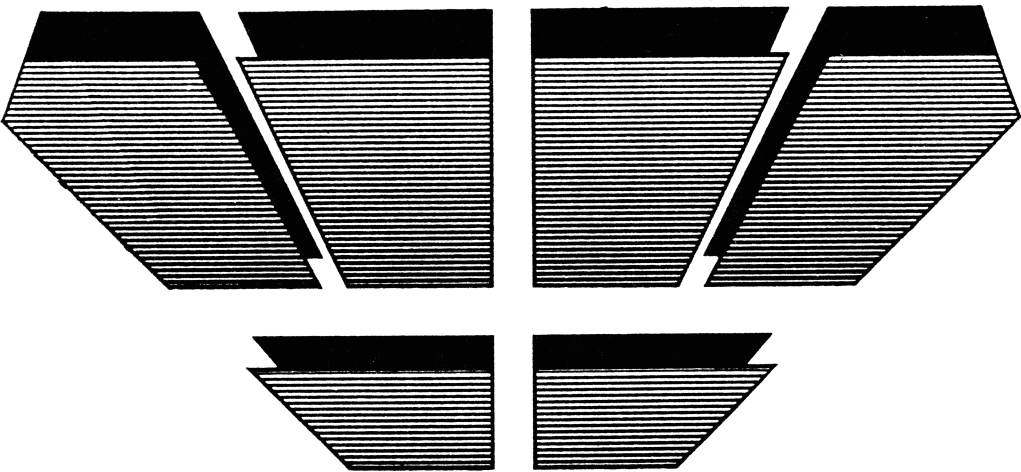


SUPER SPACE™

II



- ED/ASM MODULE
- 32 K MEMORY

MINIMUM REQUIREMENTS

- TI 99/4A Home Computer
- Extended Memory
- Disk System

SUPER SPACE I

**An 8K Byte Memory Expansion Module
With Editor/Assembler
For the TI-99/4A**

SUPER SPACE II

**A 32K Byte Memory Expansion Module
With Editor/Assembler
For the TI-99/4A**

**(c) copyright 1986
all rights reserved.**

**Module Design and Artwork (c) 1986
Diskette Program Contents (c) 1986
Manual (c) 1986
DataBioTics Inc.**

SUPER SPACE

SUPER SPACE is a cartridge-based product for the TI-99/4A from DataBioTics that is available in two models: SUPER SPACE I and SUPER SPACE II. Features common to both models will be described using the generic SUPER SPACE term. Features unique to one model or the other will be specifically indicated as such.

SUPER SPACE I consists of an 8K byte cartridge and 4 diskettes. The diskettes are the SUPER SPACE diskette, the EDITOR diskette, and 2 diskettes containing a Macro Assembler from R. A. Green. These 4 diskettes may be distributed as 2 "flippies".

SUPER SPACE II consists of a 32K byte cartridge and the same 4 diskettes as SUPER SPACE I. In addition, 2 additional diskettes (or 1 floppy) contain the c99 compiler from Clint Pulley. The SUPER SPACE II package also includes two books from Steve Davis Publishing: Programs for the TI Home Computer and Introduction to Assembly Language for the TI Home Computer.

SUPER SPACE provides the following features:

1) A total of 14K or 38K bytes of memory. This memory consists of 6K bytes of GROM and 8K or 32K bytes of RAM. The GROM is TI's standard Editor/Assembler while the RAM provides additional space for data and program storage. SUPER SPACE I provides 8K bytes of RAM while SUPER SPACE II provides 32K bytes of RAM.

2) A built-in battery in the cartridge to preserve the data or programs stored in the RAM memory.

3) Assembly language capabilities. With SUPER SPACE you can do anything you can do with the Editor/Assembler cartridge and more because SUPER SPACE provides not only the Editor/Assembler features but also adds additional RAM memory to your computer.

4) Additional TI BASIC subprograms. All the TI BASIC enhancements such as PEEK and LOAD provided by the Editor/Assembler are also included in SUPER SPACE.

5) Additional utility routines. All the program routines such as DSRLNK, GPLLNK, VSBW, and VMBW that are included with the Editor/Assembler to allow convenient interfacing to the computer's resources are also included with SUPER SPACE.

6) The ability to create your own cartridges. This manual describes how you can set up a GROM header for an assembly language program. Once your program is loaded into SUPER SPACE, it can be selected from the computer's start-up menu.

7) Sample programs and routines. The diskettes provided with SUPER SPACE contains sample programs and routines that can be used in your assembly language programs.

CONTENTS	PAGE
INTRODUCTION.....	4
SUPER SPACE HARDWARE DESCRIPTION.....	5
PARTS LAYOUT.....	5
PARTS LIST.....	6
CIRCUIT DESCRIPTION.....	7
TESTING SUPER SPACE.....	8
PROGRAMMING SUPER SPACE.....	9
USING THE RAM MEMORY.....	9
GROM HEADER 1.....	10
GROM HEADER 2.....	11
EDITOR/ASSEMBLER UTILITIES.....	12
BANK SWITCHING SUPER SPACE II.....	13
LOADING PROGRAMS INTO SUPER SPACE.....	14
DIS/FIX 80 FILES.....	14
MEMORY IMAGE (PROGRAM) FILES.....	14
GROM HEADER 3.....	15
SUPERBUG II.....	16
LINES AND CIRCLES DEMO.....	17
SUPER SPACE MENU LOADER.....	18
LOADING THE MENU.....	18
EDITING THE MENU.....	18
USING THE SUPER SPACE MENU.....	19
CARTRIDGE VACUUM/LOADER.....	21
SOFTWARE SUPPORT LOADER.....	23
LOADING THE SOFTWARE SUPPORT LOADER.....	23
CONFIGURING AN EDIT DISK.....	23
EDITOR.....	24
FORMATTER.....	24
PROGRAM EDITOR.....	24
PRINT FILE.....	25
MACRO ASSEMBLER.....	26
UTILITY.....	26
DISK DIRECTORY.....	26
PROGRAM LOADER.....	27
CONFIG PRINTER.....	28
EXIT.....	28
CUSTOMIZING THE SOFTWARE SUPPORT PROGRAM.....	28
MACRO ASSEMBLER.....	29
c99 COMPILER.....	30

INTRODUCTION

SUPER SPACE is a specially designed cartridge that provides all the features of the Editor/Assembler module plus adding 8K or 32K bytes of battery backed-up RAM memory to your computer.

This manual primarily discusses the unique features provided by SUPER SPACE. For information on using the standard Editor/Assembler features, refer to the Editor/Assembler manual (TI part number 1035984-1). Neither the Editor/Assembler manual nor the Editor/Assembler diskettes (PHD 5062) are provided with SUPER SPACE. However, a Fairware product called Macro Assembler from R. A. Green is distributed with SUPER SPACE.

A highly reliable battery back-up circuit provides long term RAM memory retention for SUPER SPACE. However, static discharge may cause data loss. To minimize this possibility, turn console power off and discharge your hand by touching some electrically grounded object before inserting or removing the cartridge.

To further safeguard your stored RAM data, avoid rapid power cycling to your console. After turning off the power to your console, wait at least 10 seconds before turning power back on. This provides adequate time for capacitors to discharge and for circuits to stabilize.

DO NOT ATTEMPT TO RECHARGE THE BATTERY. The battery supplied with SUPER SPACE is a lithium battery and is not intended for recharging. Recharging a lithium battery may be dangerous because it can cause leaking and possibly may explode. If the battery fails, throw it away and replace it with a new one. Lithium batteries such as the one used in SUPER SPACE are readily available and are very inexpensive.

The diskette entitled SUPER SPACE contains a menu loader and several assembly language source files that you can use in your programs. There is also a demonstration program that can be loaded into SUPER SPACE. The contents of the diskette are copyrighted by DataBioTics and are not to be distributed or sold except as a part of the SUPER SPACE package. You may include these source files in your original assembly language programs.

The diskette entitled EDITOR contains the revised TI Writer and Formatter files released to public domain by TI. This diskette also contains a more extensive program loader program than the version on the SUPER SPACE diskette. This loader is copyrighted by DataBioTics.

The Macro Assembler and c99 diskettes are Fairware products distributed as a courtesy to their authors. They receive no commissions or royalty from the SUPER SPACE sales so you are requested to send them a direct contribution if you find these products useful.

Each of the diskettes has a README or documentation file in DIS/VAR 80 format describing the contents of the diskette.

SUPER SPACE HARDWARE DESCRIPTION

PARTS LAYOUT

The figure below shows the parts layout for SUPER SPACE.

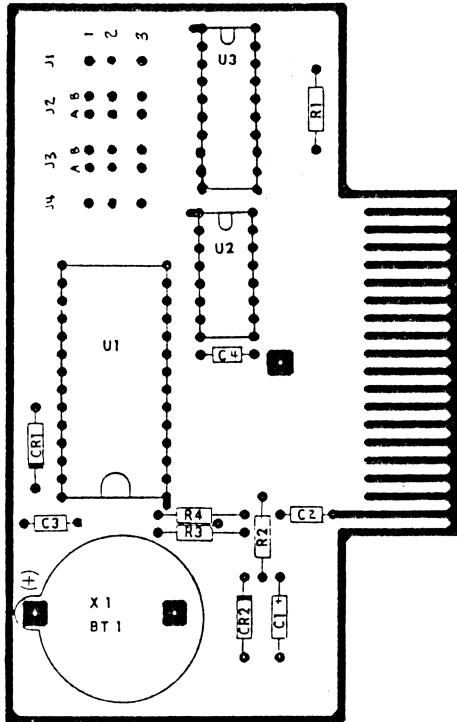


Figure 1

PARTS LIST

The following parts list describes the components that are used in SUPER SPACE.

ITEM	QTY	LOCATION	DESCRIPTION	MFG
1	1	U1 *	HM6264LP-15 8K CMOS RAM	HITACHI
		**	HM62256LP-15 32K CMOS RAM	HITACHI
2	1	U2	1015960-1204 E/A GROM	TI
3	1	U3	PAL-SSII PAL	DataBioTics
4	2	CR1,CR2	1N914 DIODE	GE
5	1	X1	BH906 BATTERY HOLDER	MPD
6	1	BT1	BR2325 LITHIUM BATTERY	MATSUSHITA
7	1	R1	100 ohm RESISTOR	DALE
8	2	R2,R3	10K ohm RESISTOR	DALE
9	1	R4	27K ohm RESISTOR	DALE
10	1	C1	22microfd CAPACITOR	SPRAGUE
11	1	C2	1 microfd CAPACITOR	SPRAGUE
12	2	C3,C4	.1 microfd CAPACITOR	SPRAGUE

NOTES:

1) The part numbers and manufacturers listed here are typical for the components used in SUPER SPACE by DataBioTics. Your particular SUPER SPACE may actually use equivalent but different parts in some cases. This list is primarily provided as a reference for spare parts purposes in case your module ever needs repair.

2) In addition to the parts listed here, a SUPER SPACE cartridge board and a TI cartridge case are also used. The SUPER SPACE cartridge boards are only available from DataBioTics. The cartridge case is the same type used with standard TI cartridges.

3) Different RAMs are used in the U1 positions for SUPER SPACE I and SUPER SPACE II. In the above parts list, the SUPER SPACE I RAM is indicated by a single asterisk (*) and the SUPER SPACE II RAM is indicated by two asterisks (**).

4) The PAL component used to perform the bank switching in SUPER SPACE II uses a proprietary logic design so this device is only available from DataBioTics.

5) The SUPER SPACE board will support 8-64K EPROM chips or 8-32K RAM chips by proper placement of jumpers in the J1-J4 positions in accordance with the following chart:

JUMPER	2764	27128	27256	27512	6264	62128	62256
J1	1-2	1-2	1-2	1-2	2-3	2-3	2-3
J2	2A-2B	2A-2B	2A-2B	3A-3B	2A-2B	2A-2B	1A-1B
J3	1A-1B	1A-1B	3A-3B	3A-3B	2A-2B	2A-2B	2A-2B
J4	1-2	2-3	2-3	2-3	1-2	2-3	2-3

Alternatively, J2 may be omitted for 6264 and 62128 chips.

CIRCUIT DESCRIPTION

The purpose of the Editor/Assembler GROM and the RAM chip are fairly evident. These are the heart of SUPER SPACE and provide its Editor/Assembler and memory expansion features.

The PAL chip contains the bank switching logic up to sixteen 8K banks of memory. It will support up to 32K bytes of RAM or EEPROM and up to 64K bytes of EPROM or ROM. It also contains logic to invert the DBIN signal for proper polarity to the OE (Output Enable) pin of the RAM chip.

Resistor R1 causes the console to reset when the cartridge is plugged in. If this resistor is removed, SUPER SPACE will work properly in every other way but it will not reset the console. Do not replace this resistor with a jumper.

Resistors R2, R3, and R4 help provide a reliable battery back-up circuit. Resistors R2 and R3 provide a pull-up to battery voltage for the RAM's WE (Write Enable) and CS1 (Chip Select 1) pins. By pulling these pins up to battery voltage, they are automatically disabled when power is removed from the cartridge, thus preventing accidental writing to the RAM during power down.

Resistor R4 in conjunction with capacitor C2 provides protection from accidental writing during power up. Resistor R4 is connected from the CS2 (Chip Select 2) pin of the RAM to +5 volts from the console. Capacitor C2 is connected from CS2 to ground. When power is first applied, this circuit keeps the RAM chip disabled for about 190 milliseconds to provide adequate time for all of the computer's circuits to stabilize. This time is short enough however that the chip can be accessed during the power-up initialization routine.

Capacitors C3 and C4 are decoupling capacitors for filtering out high frequency noise on the +5V power circuit.

The battery is a high quality lithium battery with a 160 mA (milliamp hour) rating. The shelf life of the battery is 2 to 5 years. The SUPER SPACE load is so slight that this battery should last 1 to 2 years in normal operation. If it is ever necessary to replace the battery, be sure to use one with equal specifications.

The jumpers J1-J4 are used to configure the board for the type of memory that is used. J1-2 is connected to pin 22 of the RAM or EPROM chip and connects this pin to ground or to DBIN-. J2-A1, -A2, and -A3 are connected to pin 1 of the chip and connect this pin to +5V or one of two different bank select signals. J3-A1, -A2, and -A3 are connected to pin 27 of the chip and connect this pin to +5V, WE-, or a bank select signal. J4-2 is connected to pin 26 of the chip and connects this pin to +5V or a bank select signal.

TESTING SUPER SPACE

The following steps can be used to determine if SUPER SPACE is operating properly:

1) Insert the cartridge board into the console GROM port. If this is done without the board being in the cartridge case, be sure the component side of the board is up.

Select TI BASIC from the menu screen and enter:

```
CALL LOAD(24576,1,2,3,4)
CALL PEEK(24576,A,B,C,D)
PRINT A;B;C;D
```

If you see the numbers 1 2 3 4 displayed on the screen after you type the PRINT statement, the board is working correctly.

2) Turn the console off and wait 2 to 3 minutes before turning it back on. Turn the console back on and select TI BASIC from the menu screen again. This time type only the PRINT statement. If you still get the 1 2 3 4 displayed on the screen then the battery back-up circuit is working properly.

3) If step 1 works properly but not step 2, open the cartridge case and replace the battery. Be sure the side of the battery marked + is up.

4) If step 1 does not work properly, the unit may be defective. If this is the case, try using an Editor/Assembler feature to determine if the entire unit or only the RAM is defective. If the unit is still under warranty, you should return it to DataBioTics for repair. If the warranty has expired, you may try to replace the defective component yourself or return the unit to DataBioTics for repair at a nominal cost.

5) If RAM memory does not seem to be operating properly, check the placement of the jumpers against the jumper chart to make sure your board is configured properly for the type of memory on your board.

6) If RAM memory still does not seem to be operating properly, you can try exchanging the RAM if it is plugged into a socket. Gently pry the ends of the RAM out of its socket with a thin, flat blade screwdriver and plug a replacement RAM in its place. Make sure that all pins of the replacement RAM are plugged into the socket and that the RAM is oriented in the same direction as indicated on the parts layout drawing.

PROGRAMMING SUPER SPACE

USING THE RAM MEMORY

The RAM memory in SUPER SPACE occupies the 8K byte address space from >6000 to >7FFF. (The > symbol indicates that the address value is in hexadecimal notation). The 32K bytes in SUPER SPACE II contains four 8K byte memory banks in this address space.

This memory space is directly addressable by the computer so it may be used to contain data or executable instructions for your program. If you wish to load an assembly language routine or program into this memory space, you must use an appropriate AORG instruction in your source file.

An AORG statement instructs the loader to perform an Absolute ORiGin for the program. If you do not use an AORG statement to control the loading of your program, the loader will first attempt to load the program into the first available memory from >A000 to >FFFF. If all memory in the upper 24K bytes of expansion memory is used, the loader will then try to load the program into the first available space in the lower 8K bytes of expansion memory from >2000 to >3FFF. Without an AORG statement the loader will not know the memory from >6000 to >7FFF exists.

If you only wish to use the SUPER SPACE memory for buffer storage, an AORG statement is not necessary. For example if you want to use this memory for workspace allocation, you can use LWPI instructions to specify workspace areas in SUPER SPACE. If you want to use this memory to store printer buffers or disk buffers, you can reference addresses in this space just like you would any other directly addressable memory space.

These suggestions for using SUPER SPACE only take advantage of the fact that it increases your memory expansion space from 32K to 40K bytes. While that can be important in some cases, one of the most interesting applications of SUPER SPACE is the ability to create your own cartridges.

By providing what TI calls a "GROM in ROM" header for your assembly language program, you can store a program in SUPER SPACE that can be selected from the startup menu the next time you power up your console or insert the SUPER SPACE cartridge.

The "GROM in ROM" header (or simply GROM header), is scanned by the startup program that runs when the computer is reset. The Editor/Assembler GROM is also scanned for its GROM header so when a reset occurs, the first selection will be TI BASIC the second selection will be EDITOR/ASSEMBLER, and the third selection will be taken from your SUPER SPACE header.

If you are using SUPER SPACE II, bank 0 is automatically enabled when power is applied to your console. In this mode it is functionally equivalent to SUPER SPACE I. In order to use the other 3 banks, your programs must include bank-switching routines to change the active bank.

GROM HEADER 1

The source statements below illustrate the minimum requirements for a GROM header to allow your program to be selected from the startup menu.

```

*****
*
*          SUPER SPACE GROM HEADER TEMPLATE          *
*
*****
      DEF  START1          Define label for program start
      AORG >6000          Program origin at >6000
      DATA >AA01          Verification Data (required)
      DATA 0
      DATA 0
      DATA PRGLK1          Pointer to first program linkage
      DATA 0
      DATA 0
PRGLK1 DATA 0          Pointer to next program (0 if none)
      DATA PRG1          Start address of program 1
      BYTE 11          Length of program name
      TEXT 'SUPER SPACE'    Program name to appear on menu
PRG1  B  @START1          Go to first program entry

```

This GROM header template is included on the SUPER SPACE diskette in the file called GRMHDR1. When you write your own programs to run in the SUPER SPACE, copy this template file to the same disk containing the source files for your program. In general it is recommended that your master source file be a collection of COPY directives to specify all the various source files to be used by your program.

If you want to edit the program name to something other than 'SUPER SPACE', be sure to change the BYTE 11 statement to reflect the actual length of your program's name. The first executable instruction in your program must have the label START1. If you want to use some other label, be sure to edit the DEF START1 and the B @START1 statements in the GROM header template to reflect your program's label.

If you wish to include more than one program in SUPER SPACE, the linkage for the GROM header can be extended as shown on the following page. This example is also included on the SUPER SPACE diskette in the file called GRMHDR2.

GROM HEADER 2

```

*****
*                                     *
*               SUPER SPACE GROM HEADER TEMPLATE               *
*                                     *
*****
      DEF START1      Define label for program start
      DEF START2      Define label for program start
      AORG >6000       Program origin at >6000
      DATA >AA01      Verification Data (required)
      DATA 0
      DATA 0
      DATA PRGLK1     Pointer to first program linkage
      DATA 0
      DATA 0
PRGLK1 DATA PRGLK2    Pointer to next program (0 if none)
      DATA PRG1       Start address of program 1
      BYTE 11          Length of program name
      TEXT 'SUPER SPACE' Program name to appear on menu
PRG1   B @START1       Go to first program entry
PRGLK2 DATA 0         Pointer to next program (0 if none)
      DATA PRG2       Start address of program 2
      BYTE 14          Length of program name 2
      TEXT 'SECOND PROGRAM' Program name 2
PRG2   B @START2       Go to second program entry

```

You can continue to extend this linkage scheme if you want more than 2 programs in SUPER SPACE. The key is that each PRGLK_n DATA statement points to the address, name length, and name of the next successive program and the last DATA statement in this chain must contain a 0 to indicate the end of the chain.

If you have a CorComp disk controller, only your first program name will appear on the menu screen. If you have additional programs in SUPER SPACE, press the space bar to get the color bar presentation then press any key to exit the CorComp disk controller and return to the TI console routines. At this time all of your programs (up to 7) will be presented on the menu screen.

While the CorComp disk controller's menu shows the first SUPER SPACE program from your linked list, the TI console menu shows up to 7 programs in reverse order. That is, the PRG1 program name will be at the bottom of the screen and the last program in your linked list will be option number 3 (following the Editor/Assembler selection).

The TI console menu supports up to 9 total selections, 7 of which can be SUPER SPACE programs. If you attempt to create a list of more than 7 programs, the menu screen should show the last 7 but there may also be some "garbage characters" on the screen.

Your program names should not exceed 20 characters in length. Up to 22 characters will fit on the menu line and longer names will wrap around to the next line but depending on the quality and adjustment of your monitor, the characters at the ends of the lines may be difficult to read.

EDITOR/ASSEMBLER UTILITIES

Since an Editor/Assembler GROM is included in the SUPER SPACE cartridge, all standard Editor/Assembler utilities are available when programs are written for use with the 32K memory expansion. However, if you are writing stand-alone programs that you intend to run as a "cartridge program" out of SUPER SPACE memory, then you should not use these standard utilities.

When you first turn power on your console and expansion memory, the Editor/Assembler utilities do not exist in the expansion memory. If you were to select a SUPER SPACE program that depends on these utilities, then it would not run properly. In such a case you would first have to select the Editor/Assembler option then press FCTN/= to quit and return to the menu screen and then select the SUPER SPACE program. Alternatively, you can first select TI BASIC then type CALL INIT to force the utilities to be loaded into expansion memory. After the CALL INIT, press FCTN/= to quit and return to the menu screen so you can select your SUPER SPACE program.

To eliminate the need for these extra steps, DataBioTics has included a special set of stand-alone Editor/Assembler utilities on the SUPER SPACE diskette. The source code for these utilities is in the file called UTILITIES.

The UTILITIES file contains a version of VSBW, VMBW, VSBR, VMBR, VWTR, KSCAN, GPLLNK, and DSRLNK that you can include in your original assembly language programs. The method for using these routines is the same as described in the Editor/Assembler manual for the utilities supplied with the Editor/Assembler. All of these routines must be called with a BLWP instruction and the caller's workspace registers must be set up as described in the Editor/Assembler manual.

If you use these stand-alone routines in your program, do not include any REF statements with these routine names as the operand. The REF statements cause the loader to search the REF/DEF table to find the memory addresses of the standard version of the utilities. Since the complete routine is being included in your program when using the UTILITIES file, the REF statements are not necessary.

BANK SWITCHING SUPER SPACE II

The four memory banks in SUPER SPACE II may be selected by writing the appropriate bit pattern to CRU address >0800. The following routine can be included in your assembly language programs to perform this bank switching. To use this routine, you must execute a BL @BNKSW instruction with R0 set to the desired bank number (0-3):

BNKSW	LI	R12,>0800	Set CRU Address
	LI	R1,2	Load Shift Bit
	SLA	R0,1	Align Bank Number
	SLA	R1,0	Align Shift Bit
	LDCR	R1,0	Switch Banks
	SRL	R0,1	Restore Bank Number (optional)
	RT		Return

SUPER SPACE II bank switching can also be controlled with SUPERBUG II. In this case you should use the version of SUPERBUG II that loads into high expansion memory (SBUGO). Then use the C command to select the SUPER SPACE II CRU address as follows:

C 800,8 <return>

SUPERBUG II will then attempt to read from the CRU address and will display the data (all zeroes). After this response, enter one of the following commands to perform bank switching:

BANK #	COMMAND
-----	-----
0	2
1	8
2	20
3	80

Once the desired bank has been selected, you can use the CTRL/L command of SUPERBUG II to load a program file into that bank of SUPER SPACE memory.

A selected SUPER SPACE II memory bank will remain active until power is cycled on your console at which time the unit will be automatically reset to bank 0. It is recommended that you include a valid GROM header in any program you load into SUPER SPACE memory so a menu selection can appear no matter which bank is active when the computer is reset.

The menu selection routine in the console will not perform any bank switching so only the GROM header of the currently active bank will be searched when the console is reset. If a SUPER SPACE II program is selected from the startup menu, it should perform whatever bank switching is necessary for the program to execute properly.

LOADING PROGRAMS INTO SUPER SPACE

An assembled object file (DIS/FIX 80) can be loaded into SUPER SPACE using Option 3 (LOAD AND RUN) from the Editor/Assembler. A SAVED memory image file (PROGRAM) can be loaded with Option 5 (RUN PROGRAM FILE) or with the CTRL/L option of SUPERBUG II.

DIS/FIX 80 FILES

The source file must include an AORG >6000 statement in order to force loading into SUPER SPACE memory. If the END statement has an operand, the program will auto run in the usual manner. If it does not, press ENTER to advance to the PROGRAM NAME prompt and type START1 (or whatever DEF label you used for your program). If you have included a GROM Header template, you can also select your program from the menu after resetting the console.

Once a program has been loaded into SUPER SPACE in this manner, it should be available every time you reset the console until you reload the memory with another program or until the battery runs down (1 to 2 years typically).

MEMORY IMAGE (PROGRAM) FILES

Saving and loading memory image programs is a little more difficult but since such programs load much faster than DIS/FIX 80 files, it is worth the effort. To save a file in memory image format, you will have to use the CTRL/S option of SUPERBUG II or the SAVE utility that is provided on a diskette with the Editor/Assembler. The use of the SAVE utility is described in the Editor/Assembler manual.

Before using the SAVE utility, you must assign the labels SFIRST and SLOAD to the first instruction of your program. The label SLAST must be assigned to the END statement of your program. After assembling your program with these labels, load it into memory using Option 3. Then when you are prompted for another FILE NAME, load and run the SAVE utility. Answer its prompts to save your SUPER SPACE program in memory image format.

The CTRL/S option of SUPERBUG II allows DIS/FIX 80 programs to be saved in memory image format even without the SFIRST, SLAST, and SLOAD labels.

Once your program has been saved with SUPERBUG II or with the SAVE utility, you can use Option 5 from the Editor/Assembler or CTRL/L from SUPERBUG II to load your program into SUPER SPACE. In this case, your program will automatically run the first time it is loaded. After this you can reset the console and select the program from the menu screen.

In general you should avoid the use of self-modifying code. Such a practice is prone to error and makes debugging difficult. However, this technique must be used to set up a SUPER SPACE program to be saved as a memory image. GROM Header 3 described on the following page will accomplish this.

GROM HEADER 3

```

*****
*
*
*          SUPER SPACE GROM HEADER TEMPLATE
*
*
*****
      DEF  START1      Define label for program start
      DEF  START2      Define label for program start
      AORG >6000        Program origin at >6000

SFIRST
SLOAD  B    @PATCH    Go to the patch area
*      DATA >AA01      Verification Data (comment
*      DATA 0          out the first two words)
      DATA 0
      DATA PRGLK1      Pointer to first program linkage
      DATA 0
      DATA 0

*
PATCH  LI    RO,>AA01    Get validation data
      MOV    RO,@>6000    Store validation data
      CLR    RO          Need another data value
      MOV    RO,@>6002    Store it away
      B      @START1     Go to real program entry

*
PRGLK1 DATA PRGLK2      Pointer to next program (0 if none)
      DATA PRG1         Start address of program 1
      BYTE 11            Length of program name
      TEXT 'SUPER SPACE' Program name to appear on menu
PRG1   B      @START1    Go to first program entry
PRGLK2 DATA 0           Pointer to next program (0 if none)
      DATA PRG2         Start address of program 2
      BYTE 14            Length of program name 2
      TEXT 'SECOND PROGRAM' Program name 2
PRG2   B      @START2    Go to second program entry

```

This GROM header is included in the SUPER SPACE diskette in the file called GRMHDR3. This file should be used instead of GRMHDR1 or GRMHDR2 if you are writing a program that you intend to convert into memory image ("PROGRAM") format for loading into SUPER SPACE.

When a program with this GROM header is loaded via Option 5 of the Editor/Assembler or with the CTRL/L option of SUPERBUG II, it will start running at location >6000. The first instruction is a branch to the patch (self-modifying code) routine. The patch routine then stores the values >AA01 and >0000 in locations >6000 and >6002. This overlays the branch instruction with values the start-up routine expects to find in a GROM header. After patching these two locations, the routine branches to the location DEFINED as the first program.

Be sure to include the label SLAST on the END statement of your program if you want to use the SAVE utility to generate a memory image file.

SUPERBUG II

Edgar Dohmann, the designer of SUPER SPACE, is offering SUPERBUG II as a FAIRWARE product. SUPERBUG II is a vastly improved version of the SUPER BUGGER debugging program which TI released to public domain. The SUPERBUG II diskette contains a program file that will load into SUPER SPACE in addition to a standard DIS/FIX 80 file version and a version that is fast and convenient to load in an Extended Basic environment.

SUPERBUG II contains the following enhancements:

- 1) Bugs that affected memory dumps and disassembly to disk as well as several other bugs have been fixed.
- 2) A faster loading and more convenient to use version for Extended Basic is included.
- 3) An O option has been added to allow the output device to be changed. The output device is opened with an APPEND specification rather than an OUTPUT to avoid overwrites.
- 4) The C, F, G, H, K, N, >, and . commands from TI DEBUGGER are included in SUPERBUG II.
- 5) Disassemblies can be done as DATA statements or as instruction mnemonics. Address and instruction values can be included or omitted for either screen, printer, or disk output. A relocation feature is included in case the code to be disassembled was copied from another location (such as a DSR ROM).
- 6) Relocations of data blocks can be performed in normal or slow speed (10 to 20 milliseconds per byte) so SUPERBUG II can be used to copy data into EEPROM devices.
- 7) An option has been added that allows toggling of screen colors.
- 8) The CTRL/L and CTRL/S options have been added to allow loading and saving program files from SUPERBUG II.
- 9) A 52-page manual describing SUPERBUG II is included with each order.

To order your copy of SUPERBUG II, send \$10.00 to cover the cost of diskette, manual, mailer, postage, and copying to the author at the address below:

Edgar L. Dohmann
Rt. 5 Box 84
Alvin, TX 77511

If you obtain a copy of SUPERBUG II through public domain distribution, you are requested to send \$5.00 to the author. In return, you will be sent a printed copy of the manual.

LINES AND CIRCLES DEMO

The SUPER SPACE diskette contains the source and object files for a bitmap graphics demonstration program. The program will draw a sequence of lines or circles on the screen and includes an option to print the current screen.

The master source file is called DEMOSRC. This file contains the GROM header which allows the LINES DEMO or the CIRCLES DEMO to be selected from the startup menu. This file also contains COPY directives for the UTILITIES file and the LCPSRC file.

The LCPSRC file contains the main program code for the Lines, Circles, and Print routines. The source for these routines is provided to allow you to customize the demo to suit your own needs and to help provide some insight into bitmap graphics programming. You may also include these routines into other programs that you write.

The object file is called DEMO and should be loaded with Option 3 of the Editor/Assembler. Once the program is loaded, press ENTER to advance to the Program Name prompt and type LINES or CIRCLS to start either the Lines or the Circles demo. Once the computer is reset, these demos can be started by a single key selection from the startup menu.

While the demo program is running the following keys can be pressed to change the operation of the program:

- O changes to circle mode
- L changes to line mode
- P prints the current screen to a graphics mode printer
- C toggles the color-freeze/multi-color mode of operation
- Q quits the program and resets the computer

The print routine in the LCPSRC file is configured for a Texas Instruments Model 855 printer operating serially at 9600 baud on RS232 port 1. If your printer requires a different configuration, change the printer description in the PABC data block near the end of the LCPSRC file.

Other areas of the LCPSRC file that may require modification for your printer's characteristics include the following:

- 1) ONGRPH data block -- This data block contains the control codes to set the vertical line length at 8/72 and the horizontal graphics mode at 72 dots/inch, 256 dots/line.
- 2) CRLF data block -- This data block is sent at the end of each 256 bytes of graphics data to cause the printer to print the current contents of its line buffer.
- 3) SIXLPI data block -- This data block is sent at the end of the screen dump to return the printer to its normal six line per inch vertical line spacing.

SUPER SPACE MENU LOADER

The SUPER SPACE diskette contains two files which provide the capability of storing a menu and loader in the SUPER SPACE memory area that will auto-load and run assembly language programs. MBAS is a BASIC program that will load a menu template and allow you to edit the menu selections. MOBJ is the menu template that is loaded into SUPER SPACE.

LOADING THE MENU

With the SUPER SPACE cartridge installed, load and run the MBAS program in the same way you would any BASIC program. You will be presented with a title screen containing 4 choices as follows:

PRESS	FOR
1	LOAD MENU TEMPLATE
2	INIT MENU TEMPLATE
3	EDIT MENU TEMPLATE
4	EXIT PROGRAM

Selecting the first choice will cause the Menu Template to be loaded into SUPER SPACE memory. The disk containing the MOBJ file must be named SUPERSPACE when this is done. After loading is complete, you will be returned to the title screen.

Selecting the second choice will initialize the Menu Template. This is not necessary when first loading and editing the menu because the MOBJ file contains an initialized menu. This choice should be used when you want to delete some of the menu entries. It will cause all entries to be deleted so you can enter a fresh set of programs. After initialization is complete, you will be returned to the title screen.

The third choice allows you to edit the menu template by entering file and program names that you wish to be able to load from the startup menu screen.

The fourth choice exits the program and returns to the BASIC command mode.

EDITING THE MENU

When the third choice is selected from the title screen, you will be presented with the SUPER SPACE Edit Screen. This screen has headers for PGM #, DSK #, FILE NAME, FILE TYPE, and ENTRY NAME. Up to 7 programs may be entered with this Edit Screen. Before the cursor is presented and you start entering information, the program will scan the SUPER SPACE RAM and fill in any programs that are already defined.

The cursor first appears in the first column under PGM #. Programs should be entered in order from 1 to 7 to ensure that linkage is set up properly. You may use the FCTN/X and FCTN/E keys to move the cursor up and down this column to edit different lines. The FCTN/S and FCTN/D keys are also operational to move back and forth on a line. The FCTN/I key may be used to delete characters in the FILE NAME and ENTRY NAME fields while the FCTN/3 key may be used to erase a FILE NAME or ENTRY NAME while in one of those fields.

The ENTER key is used to move the cursor from field to field if less than the maximum allowed characters are entered. File names may be up to 10 characters in length while Entry names may be up to 6 characters long. Once the ENTER key is pressed after entering an Entry name, the program will place the data entered for that line into SUPER SPACE RAM.

The FCTN/9 key will terminate edit mode and return to the main menu screen. The FCTN/= key will exit the program and reset the computer.

The DSK # entry must be a value from 1 to 6 or an X. An X is allowed for compatibility with Foundation 128K cards with disk emulation.

The FILE TYPE entry must be a 0 if the file is a DIS/FIX 80 type of assembly language file or 1 if the file is a PROGRAM type. The DIS/FIX 80 files are those normally loaded with Option 3 (LOAD AND RUN) of the Editor/Assembler and PROGRAM files are those normally loaded with Option 5 (RUN PROGRAM FILE).

If you entered 1 for the FILE TYPE, all information is complete and you may leave the ENTRY NAME blank. However, if you entered 0, you must enter an ENTRY NAME unless the DIS/FIX 80 file is set up to auto-run when loaded in which case, you may leave the ENTRY NAME blank. The ENTRY NAME must be the same program name that you would type to start the program after loading it with the Editor/Assembler Option 3.

If a FILE NAME is blanked, the remaining information on that line will remain in SUPER SPACE RAM, but the linkage to that program and any following programs will be deleted and only the programs on lines above the blank File name will appear on the startup menu. To restore a deleted linkage like this, it is only necessary to re-enter a File name.

USING THE SUPER SPACE MENU

Once you have entered the information for 1 to 7 programs using the Edit option as described above, you are ready to use SUPER SPACE to auto-load and run your programs. After exiting the MBAS program, reset your computer by cycling power on the console or pressing the reset switch if you have a cartridge expander. If you have to cycle power to reset your computer, leave the power off for at least 10 seconds before turning it back on to minimize the chance of data loss during power cycles.

After restarting your computer, your startup menu screen should show the file names of the programs you entered into the menu template. To run one of your programs, first make sure you have the disk containing the assembly language program file in the drive you specified when entering the information, then press the number indicated by the selection to load and run the program.

If an error occurs in loading or running the program, the computer will be reset back to the power up screen. If this should occur, repeat your editing steps for the program and make sure you are using the right disk drive number, file name, and program name. If this is all correct, make sure it is possible to load the program by using either Option 3 or Option 5 of the Editor/Assembler cartridge. If there is a file error or a data format error in the file, an error message will be displayed when you try to load it from the Editor/Assembler.

If you have a CorComp disk controller card, only the first program selection you entered will show up on the initial startup menu screen. If you have entered more than one program, press the space bar twice to exit the CorComp screen and obtain a normal TI startup menu. At this time you will be presented with all of your entries.

Unless a static discharge disrupts the contents of your SUPER SPACE memory, the menu information you have entered should be available every time you power-up or reset your computer with SUPER SPACE installed. This information should remain in SUPER SPACE until you edit the program information or load some other program into SUPER SPACE.

In general, you should only use this feature of SUPER SPACE to load and run programs that reside in the 32K expansion memory space from >2000 to >3FFF and >A000 to >FFFF. The menu template and loader programs for this feature all reside in SUPER SPACE memory and any attempt to load programs into the memory space from >6000 to >7FFF will probably not work. The memory space from >6900 to >7C00 is available for your use with Version 1.0 of this program but continued availability of this space is not guaranteed for future releases of this program.

CARTRIDGE VACUUM/LOADER

The SUPER SPACE diskette contains an assembly language program called CVAC which allows you to save ROM-only cartridges to disk and reload the files into SUPER SPACE for execution.

CVAC must be loaded with Option 3 of the Editor/Assembler. After loading is complete, you must press ENTER and advance to the Program Name prompt then type CVAC before the program will run. The program does not auto run so you can load additional programs into Expansion Memory after CVAC (such as SUPERBUG II) if desired.

When CVAC is started, you will be presented with a menu screen of 4 selections as follow:

PRESS	FOR
1	COPY A CARTRIDGE TO DISK
2	LOAD A CARTRIDGE FILE
3	RUN A PROGRAM
4	RESET COMPUTER

Selection 1 will first prompt you for a File Name. In response to this prompt, type the file path name where you want the cartridge image to be stored. The program will then read the contents of a cartridge RAM or ROM from >6000 to >7FFF and store this image in PROGRAM format in the file you specified. If an error is detected, a message will be displayed on the screen indicating an I/O Error has occurred. After the cartridge contents are saved, the program will return to the main menu.

If you are saving the contents of some other cartridge besides SUPER SPACE, you will need to switch cartridges after loading CVAC and before entering the File Name after selecting Option 1. The easiest way to do this is to use a Cartridge Expander device which allows you to switch cartridges without resetting the computer. If you do not have a Cartridge Expander, you must devise a manner of switching cartridges without resetting the computer. One possibility is to place a small piece of cellophane tape over the reset pin (pin 1) of the cartridge you wish to save.

Option 2 also prompts for a File Name, then loads the specified file into SUPER SPACE RAM. If an error is detected, a message will be displayed on the screen indicating that an I/O Error has occurred. After loading is complete, the program will return to the main menu.

This feature should only be used to load PROGRAM files that were created by the CVAC program. These files have a slightly different disk storage format than files which are saved by the Editor/Assembler SAVE utility. Hence, this feature is not compatible with Editor/Assembler Option 5 and vice versa.

Option 3 will prompt you for a program name to be executed. This is allows you to exit CVAC and run another program (such as a debugging program like SUPERBUG II). This provides a convenient means of editing, disassembling, or debugging a cartridge file after loading it into SUPER SPACE.

The recommended approach is to first load CVAC then the debugging program. This will assure that CVAC is loaded starting at >A000 and the debugger will load in the first available location after CVAC. When the debugger is loaded, its name (SBUG for SUPERBUG II) will be placed into the REF/DEF table which CVAC searches after you type a Program Name in response to CVAC Option 3. If the program name you specified is not in the REF/DEF table, an Error message stating Program Not Found will be presented.

If the CVAC program is loaded first, its entry point will be at >A000. This provides a convenient means of exiting the debugger and returning to CVAC. With SUPERBUG II, use the R command to set a Workspace at >F000 (or some other unused area) and a Program Counter Address of >A000, then use the E command to execute CVAC.

Option 4 will exit CVAC and reset the computer so the cartridge program loaded into SUPER SPACE can be selected from the start-up menu.

NOTICE:

THIS FEATURE IS PROVIDED BY DataBioTics, Inc. AS A MEANS FOR BACKING UP YOUR ROM-ONLY CARTRIDGES TO DISK. THIS FEATURE IS NOT TO BE USED TO MAKE DISK COPIES OF CARTRIDGES THAT YOU DO NOT OWN. DataBioTics, Inc. DOES NOT CONDONE THE USE OF THIS PROGRAM OR ANY OTHER SUPER SPACE FEATURES FOR ILLEGAL ACTIVITIES.

CVAC is intended to be used only with ROM-only cartridges. It does not make a disk copy of any GROM code that may be present in the cartridge. CVAC also only copies 8K bytes of memory from a cartridge to disk; therefore, it will not work on any cartridges which use a paging technique to access more than 8K bytes of memory.

SOFTWARE SUPPORT LOADER

The EDITOR diskette contains a file called SSLDR which is an assembly language program that allows a number of Software Support Options to be conveniently loaded and executed. The source files (SSLDRC and SSLDRA through SSLDRF) are also included on this diskette so you can customize it as needed.

LOADING THE SOFTWARE SUPPORT LOADER

SSLDR is an assembly language program file, so it must be loaded with Editor/Assembler Option 5. It will load into SUPER SPACE memory and will automatically run when loading is complete. Once the program is loaded into SUPER SPACE memory, it will appear on the startup menu as SOFTWARE SUPPORT when the console is reset.

SSLDR will present a menu of 10 choices as follows:

- 1 Editor
- 2 Formatter
- 3 Program Editor
- 4 Print File
- 5 Macro Assembler
- 6 Utility
- 7 Disk Directory
- 8 Program Loader
- 9 Config Printer
- 0 Exit

Pressing the number before each of the menu choices will cause that selection to be executed. CTRL/3 may be used to toggle through a set of screen color combinations. Each of the choices is described in detail in the following sections. In order for some of the choices to operate properly, you must first configure a backup diskette named EDIT.

CONFIGURING AN EDIT DISK

In order to get the maximum benefit out of the SOFTWARE SUPPORT LOADER, you should first format a diskette and give it the name EDIT using your favorite Disk Manager. Once the diskette has been formatted, copy the files SSLDR, EDITB1, EDITB2, FORMB1, FORMB2, and PMENU from the EDITOR diskette to your new diskette named EDIT. If you want to use the Macro Assembler, copy the files RAGASM and RAGASN from MACRO ASSEMBLER diskette 1 to your new diskette named EDIT.

This new EDIT diskette should be used when you select options from the SOFTWARE SUPPORT screen. In the following sections, additional information will be given on how you can further expand the contents of your EDIT diskette.

The EDIT diskette can be placed into any drive on your system.

EDITOR

The Editor option on the SOFTWARE SUPPORT screen loads and runs the files EDITB1 and EDITB2 on your EDIT diskette. These files contain slight modifications to the improved TI Writer Editor that was released to public domain by Texas Instruments.

EDITB1 and EDITB2 have been modified to use the SOFTWARE SUPPORT Disk Directory feature if the SD (Show Directory) command is used while in the Editor. An exit from the Editor will return to the SOFTWARE SUPPORT menu.

A TI Writer manual is not included with SUPER SPACE but there is some reference information in the README file on the EDITOR diskette. If you need a TI Writer manual, you should be able to purchase one from Texas Instruments for a nominal fee. Request TI Writer Manual Number 1053597-1 when you contact them. Dr. William Browning's publication, "TI-WRITER COMPANION", is also an excellent reference document.

The Editor is intended for word processing applications. It will generate DIS/VAR 80 text files that are compatible with the Formatter. In general, this Editor selection should not be used to generate assembler or c99 program source files.

FORMATTER

The Formatter option on the SOFTWARE SUPPORT screen loads and runs the files FORMB1 and FORMB2 on your EDIT diskette. These files contain slight modifications to the improved TI Writer Formatter that was released to public domain by Texas Instruments.

FORMB1 and FORMB2 have been modified to exit to the SOFTWARE SUPPORT menu.

No documentation is provided with SUPER SPACE on the TI Writer Formatter. If you are not familiar with this formatter, you should obtain a copy of the TI Writer manual from Texas Instruments.

The Formatter is intended for printing files generated by the Editor. In general it should not be used to print assembly language or c99 program source or listing files.

PROGRAM EDITOR

The Program Editor option on the SOFTWARE SUPPORT screen loads and runs the files EDITB1 and EDITB2 on your EDIT diskette. Before running the program however, some additional dynamic patches are made to the memory image of the program.

Tab settings are changed to be more compatible with assembly language programming, word wrap is turned off, and fixed mode is set. These changes automatically put the TI Writer Editor into a configuration that is suitable for creating assembly language and c99 program source files.

The TI Writer manual explains the unique features of the TI Writer Editor when operating in fixed mode. Essentially this places it in a mode that is compatible with the editor that is supplied in the Editor/Assembler package.

This Program Editor has some advantages over the Editor/Assembler editor however. Since all of its commands are fully compatible with the TI Writer Editor, there is no confusion involved when going from one editor to another. Saving files is easier and more convenient with the Program Editor because it can be done from the command line rather than going back to a menu and it remembers the last file that was loaded.

PRINT FILE

This option allows you to print a DIS/VAR 80 text file. It is intended primarily to print files generated by the Program Editor but it can be used to print any DIS/VAR 80 file.

When this option is selected, the screen will be cleared and a file name will be shown in the upper left corner of the screen. If you want to print a different file, merely type the name of the file you wish to print. No cursor is shown, but the characters you type will replace those on the screen. When you have typed your new file name, press ENTER and the file will be printed. If your file name is shorter than the one initially shown, the extra characters in the old name will be ignored.

If you make a mistake while entering the file name, press FCTN/S for a backspace and you can start over. If you want to print the file initially shown, merely press ENTER. If you decide not to print a file after getting to the file name screen, press FCTN/9 to return back to the SOFTWARE SUPPORT menu screen.

Files can be printed from the Editor or Program Editor but these programs require that the file first be loaded into memory. The Print File option allows files to be printed without this requirement. This is sometimes more convenient and allows larger DIS/VAR 80 files to be printed because memory size limitations is not a consideration.

Files can also be printed by selecting Editor/Assembler from the startup menu screen then selecting the Edit Option and then selecting the Print option. The SOFTWARE SUPPORT Print File option works similar to this Editor/Assembler print option but is sometimes more convenient because an exit back to the startup menu screen is not necessary if you are operating in the SOFTWARE SUPPORT environment.

MACRO ASSEMBLER

This option will load and run the Macro Assembler if you have placed the files RAGASM and RAGASN on your EDIT disk. When the Macro Assembler starts, you will be presented with a screen prompting you for file names needed in order for the assembler to execute.

The Macro Assembler operation is documented on MACRO ASSEMBLER distribution diskette 1. You should use the Formatter option to print file DOCASM to get a printout of this file.

UTILITY

This option will search disk drives 1-6 for a file called UTIL1. If it finds a file with that name, it will load and execute that file. This provides a feature similar to the Utility option on the TI Writer cartridge.

You can place an assembly language program file named UTIL1 on your EDIT diskette if you wish and it will be loaded and executed when this option is selected.

An excellent program called 99/4A AUTO SPELL-CHECK by Dragon-slayer American Software Company uses a file called UTIL1 to allow this spelling checker to be loaded from the TI Writer cartridge. If you own this program or another with such a load feature, it can also be loaded with this Utility option on the SOFTWARE SUPPORT screen.

DISK DIRECTORY

This option presents a disk directory on your monitor screen. When the option is first selected, it will prompt you for a disk number by displaying DSK? in the upper left corner of the screen. Type the drive number from 1 to 6 that you wish to catalog.

The catalog on the specified drive will be presented in two columns with 20 files per column. If the diskette contains more than 40 files, a message will be displayed at the bottom of the screen instructing you to press any key for the next page. If the catalog is complete, a message will be presented requiring a Y entry to catalog another diskette or an N entry to return to the SOFTWARE SUPPORT menu.

When a catalog is presented on the screen, you may print the screen by pressing FCTN/0. The last line containing the prompt instructions will not be printed. The printout will be sent to the default device name or the name last entered with the Config Printer option.

The disk directory is presented in two columns with up to 20 file names per column. The 40-column text mode is used for this display. Ten spaces are used for the file names, 4 spaces are used for the file size, and 5 spaces are used for the file attributes. The center 2 spaces of the screen are left blank as a separator between the two columns of file names.

The first space of the file attributes is blank if the file is unprotected or a small p is displayed if the file is protected. After the protection indicator, the file type is shown. Program files are abbreviated as Prog while other file types are indicated by a letter and record size. The letter used is d for DIS/VAR, D for DIS/FIX, i for INT/VAR, or I for INT/FIX. The record size is indicated in the three spaces following the letter.

This same disk directory is used when the SD command is executed from the Editor or the Program Editor. In this case, the disk number is prompted in the editor rather than on a separate screen and an exit from the disk directory will return you to the editor rather than to the SOFTWARE SUPPORT screen. The print screen feature will also work when the SD command is executed.

PROGRAM LOADER

This option will search disk drives 1-6 for a file called PMENU. If such a file is found, it will be read by a routine in the SOFTWARE SUPPORT program to present an alternate program menu display. In this case the program names will be preceded by a letter that must be pressed in order to load and execute the program.

Use the Program Editor to examine the format of the sample file called PMENU on the EDITOR disk which you may have copied onto your EDIT disk. One program name must be placed on each line and the program name must be followed by the file name to be loaded.

Program names may be up to 12 characters long and may include imbedded blanks. The file pathname must start in column 13. Winchester disk directory pathnames are supported.

This feature only loads PROGRAM files. If you have a DIS/FIX 80 assembly language program that you wish to load, you should first convert it to Program format using the SAVE utility from the Editor/Assembler package or the CTRL/S command from SUPERBUG II. These utilities are not included with SUPER SPACE.

CONFIG PRINTER

This option lets you change the default printer name used by the Print File option and the catalog print feature. The original printer name in the SSLDR source file is RS232.BA=9600 and if your printer name is different, you should edit this source file and reassemble the SSLDR program.

The Config Printer option allows you to temporarily change the default printer name in memory. When this option is selected, the screen will be cleared and the current printer name will be shown in the upper left corner of the screen. Type your new printer (or file) name over the one shown just as is described for the Print File option.

The FCTN/S, FCTN/9, and ENTER keys work the same as described for the Print File option. After pressing ENTER, you will merely be returned to the SOFTWARE SUPPORT menu and the device name you entered will be used the next time you select Print File or print a disk directory.

EXIT

This option exits the SOFTWARE SUPPORT program and returns you to the startup menu screen.

CUSTOMIZING THE SOFTWARE SUPPORT PROGRAM

The source files for the SOFTWARE SUPPORT program are included on the EDITOR diskette. The control file for the source is called SSLDRSRC. The files which actually contain the executable code statements are SSLDRA through SSLDRF.

You should use the Program Editor feature to make your changes to the appropriate files. After your changes are made, execute the Macro Assembler or the TI Assembler, whichever you prefer. You should enter SSLDRSRC as the source control file and direct the object output to SSLDROBJ.

The SSLDROBJ file will be in DIS/FIX 80 format and can be loaded with Editor/Assembler Option 3. After it is loaded you can press ENTER to advance to the Program Name prompt. At this time you should type SFIRST or SLOAD to execute the program and present the SOFTWARE SUPPORT menu.

It is recommended that you save the contents of SUPER SPACE memory to a file called SSLDR to replace the original version with your new version in a PROGRAM file format. You can use the SAVE utility in the Editor/Assembler package or the CTRL/S feature of SUPERBUG II for this purpose. The advantage of saving your new version in a PROGRAM file format is that it will load faster and auto-run.

MACRO ASSEMBLER

The Macro Assembler included with SUPER SPACE I and SUPER SPACE II is a FAIRWARE product by R. A. Green. This product is distributed with the author's permission with no royalties or commissions being paid to the author. If you like the Macro Assembler package, you are requested to send a donation directly to the author at the following address:

R. A. Green
1032 Chantenay Drive
Gloucester, Ontario
Canada K1C 2K9

The documentation for the Macro Assembler is contained on diskette 1 of the Macro Assembler package in the files called DOCASM and DOCASM1 through DOCASM3. You can use the Formatter to print DOCASM and it will automatically chain print the remaining files. The complete documentation package is 18 pages long.

The Macro Assembler is compatible with the TI Assembler except for some instructions and directives not supported by the TI-99/4A. In addition to a macro facility, the Macro Assembler also offers an improved listing format, an improved diagnostic format, and a cross-reference listing.

The Macro Assembler package includes a library of general purpose macros that will help reduce the number of lines of code that must be typed in your programs. Also included are subroutines for keyboard, VDP, and sequential file I/O as well as a GPLLNK routine.

Diskette 2 of the Macro Assembler package contains the source files for two complete sample programs which you can assemble with the Macro Assembler. One of these programs is a disk catalog program that will list the catalogs of up to 255 disks and the other is a terminal emulator program.

DataBioTics wishes to thank Mr. Green for granting us permission to distribute his excellent program with SUPER SPACE I and SUPER SPACE II. This is an excellent product that has required a lot of the author's time and effort to develop. If you like the product and use it, you are urged to send a reasonable contribution to the author at the address given above.

c99 COMPILER

The c99 Compiler included with SUPER SPACE II is a FAIRWARE product by Clint Pulley. This product is distributed with the author's permission with no royalties or commissions being paid to the author. If you like the c99 package, you are requested to send a donation directly to the author at the following address:

Clint Pulley
38 Townsend Avenue
Burlington, Ontario
Canada L7T 1Y6

The documentation for the c99 Compiler is contained on the diskettes containing the c99 Compiler. Diskette 1 contains the c99 Compiler itself while diskette 2 contains some support routines and sample programs.

The main documentation for the c99 Compiler is contained in the files C99MAN1 through C99MAN3 on diskette 1. These files can be printed with the Print File option of the SOFTWARE SUPPORT program. The total length of these files is 15 pages. Diskette 1 also contains a file called -README1 which contains 2 pages of release note documentation on Ver 2.0 of the c99 Compiler. Also on diskette 1 is a file called C99SPECS which includes 4 pages of c99 specifications.

Diskette 2 contains a file called -README2 which contains 2 pages of release notes for the support and sample programs on this diskette. This diskette contains bitmap graphics, floating point, speech, sound, and random number support routines.

The support routines all have a documentation file that is easily identified by the letters DOC or DOCS. These include PRINTDOC, BITDOC, FLOATDOC, RUNOFFDOC, GRFLDOCS, SPEECHDOCS, and TCIODOC. These files can also be printed with the Print File option.

DataBioTics wishes to thank Mr. Pulley for granting us permission to distribute this c99 Compiler with SUPER SPACE II. The author has spent substantial time and effort in developing this product and you are urged to support him if you find this program useful to you. Please send your contributions directly to the author at the address given above.

WARRANTY COVERAGE

WARRANTY DISCLAIMERS

Some states do not allow the exclusion or limitation of implied warranties or consequential damages, so the above limitation or exclusion may not apply to you in those states.

This warranty gives you specific legal rights, and you may also have other rights that vary from state to state.

DataBioTics Inc. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material or product described by this manual.

This manual contains information protected by copyright. All rights are reserved. No part of this manual may be photocopied or reproduced in any form without prior written consent from DataBioTics Inc.

