

ASGARD EXPANDED MEMORY SYSTEM

AEMS USER'S GUIDE

=====	SW 99ers	User's
=====	Asgard	Guide
=====	Memory	Version 2.0
== SAMS ==	System	R. A. Green
=====		
=====		

The Asgard Expanded Memory System, for which this software was originally developed, and after which it was named, was never built. Instead a simpler memory card called the Asgard Memory System (AMS) was built. The South west 99ers group built a better version of this card called the Super AMS. This software retains the AEMS name, but supports both the AMS and SAMS Cards.

Both these expansion memory cards provide the TI 99/4A User with a large paged memory. The software supplied with the SAMS and that available separately make the use of the large memory easy for both the user of the system and the programmer. This manual describes how to BOOT the system software and how to use the system MENU.

CONTENTS

INTRODUCTION	1
DISK CONTENTS	1
BOOTING THE AEMS SYSTEM	1
RESTARTING THE AEMS SYSTEM	2
USING A SUPER E/A CARTRIDGE	2
THE MAIN MENU	2
TAILORING THE MAIN MENU	4
MODIFYING PROGRAMS	6
OTHER SOFTWARE AVAILABLE	7

INTRODUCTION

The Asgard Memory System (AMS) and the Super AMS (SAMS) provide the TI 99/4A with lots of memory. The memory is "paged" into the normal 32K RAM addressing space in 4K blocks. The AEMS system software provides both the user and the programmer easy access to all of the RAM on the Card. Some of the features of the system software are:

1. Provides a selection menu for quick and easy access to your E/A programs on disk.
2. Allows well behaved, restartable programs to be made resident in memory for a session. This gives instant access to these programs from the main menu.
3. Provides a large subroutine library that allows Assembler Language programmers easy use of the large memory.
4. The SAMS card maps all 8 pages in the TI 99/4A address space. The AMS card maps only the 6 pages in the high memory area.

DISK CONTENTS

The distribution disk is a double sided single density disk containing the following files.

ABOOT	The AEMS Initial Load Program.
AEMSDOC	The documentation for the AEMS System, to be printed by the TI WRITER formatter.
AEMSRES1	Resident code segment 1.
AEMSRES2	Resident code segment 2.
ASEA	Super E/A Cart version of ABOOT and ASHOE.
ABOOTMODS	Patch file to modify device and file names for ABOOT.
ASHOE	The AEMS Restart Program.
TIWFORM	Version of TI Writer Formatter to print the documentation file.
XBOOT	The AEMS XB Initial Load Program.
XSHOE	The AEMS XB Restart Program.
ZAEMSPAT/D	Documentation for ZAEMSPAT.
ZAEMSPAT	Public Domain program file patcher.
ZXBPATCH	XB loader for ZAEMSPAT.

BOOTING THE AEMS SYSTEM

Getting started with the AEMS system is easy, but first, MAKE A WORKING COPY OF THE DISTRIBUTION DISK.

The distribution disk is ready to go. Later you will want to tailor the main menu for your own programs, but right away you can place your working copy of the distribution disk into DSK1 and run the E/A Option 5 program, ABOOT, or the XB program, XBOOT, to see how the main menu looks.

The patch file "ABOOTMODS" can be modified with any editor and used with the Patch program, AEMSPAT, to modify the device and file names that ABOOT uses. Comments in the patch file show how to make changes.

RESTARTING THE AEMS SYSTEM

When the AEMS system is exited, by selecting item D from the main memory, or by running a program that exits badly, the system can be restarted by executing the ASHOE program (E/A Option 5). ASHOE is a small program that checks for the AEMS system being loaded, and if so exits to it. If AEMS is not loaded when ASHOE is run it exits to the TI Colour Bar.

USING A SUPER E/A CARTRIDGE

Program ASEA is designed for an E/A Super cartridge (an E/A cartridge with RAM). When executed, using E/A Option 5, it leaves itself in the Cartridge RAM and provides two entries on the TI Main Menu that allows you to easily execute either ABOOT or ASHOE.

THE MAIN MENU

This section describes the main selection menu provided by AEMS. The menu is a double column list with items from A to V. The first four items on the menu are fixed. They provide utility functions:

- A Load and Run.
- B Load and Retain.
- C Memory Status.
- D Exit to TI Menu.

The other entries are provided by you. They represent E/A programs to be loaded and run from disk or that have previously been loaded into memory. A program or utility function is selected by pressing the corresponding letter key in either upper or lower case. Most programs will return to (or can be made to return to) the main menu display ready for the next selection.

Directory Aid

While the main menu is displayed or when a user input is requested, pressing AID (FCTN 7) opens a "Directory Aid" dialog box that allows you to display floppy or hard disk directories and to make a file name selection. In the Directory Aid dialog box, the disk device name is entered in the usual way, including

hard disk sub-directories, with or without the trailing period. The file name is selected by scrolling the cursor up and down and then pressing ENTER. Pressing BACK cancels the directory aid without making a selection.

Load and Run

Selecting item A, "Load and Run", from the main menu, allows you to run any E/A Option 5 or AEMS program file from disk. A dialog box is opened requesting the program file name. During data entry in the dialog box the function keys perform as ordinarily defined by TI. In particular,

FCTN 1 (DEL)	Delete character,
FCTN 2 (INS)	Insert character,
FCTN 3 (ERASE)	Erase to end of field,
FCTN 4 (CLEAR)	Erase entire field,
FCTN 5 (BEGIN)	Begin execution of function,.
FCTN 6 (PROCD)	Proceed with function,
FCTN 7 (AID)	File name selection from directory,
FCTN 8 (REDO)	Redo data in field,
FCTN 9 (BACK)	Terminate function,
FCTN = (QUIT)	Quit Library Manager,
ENTER	Move cursor to next field,
FCTN E (Up)	Move cursor to previous field,
FCTN X (Down)	Move cursor to next field,
FCTN S (Left)	Move cursor left,
FCTN D (Right)	Move cursor right.

When ENTER or FCTN X is pressed for the last field, the function is executed.

The dialog box may display an error message. Pressing any key clears the message and returns to the main menu display.

Load and Retain

Selecting item B, "Load and Retain" allows you to load a new program into memory, adding it to the main menu for later instant access. A dialog box is opened requesting the program's file name and the menu text to be used for the program. For a program to be resident it must be well behaved and be restartable. Not all programs behave this way. You can find out if a program is a candidate for being made resident by trying it.

Memory Status

Selecting item C, "Memory Status" gives a short display of how the AEMS memory pages are being used, and a menu of functions. The three functions on the menu allow you to "free up" memory pages for that "big job". When the pages used by retained programs are freed, the menu items for these programs will

disappear. Note that after the reserved pages have been freed you should not "Load and Retain" other programs.

In the memory status display: "Available" shows the number of pages available for user programs; "System" shows the number of pages used by or reserved for the AEMS system code; "Retained" shows the number of pages occupied by programs retained in memory ready to run; "Reserved" shows the number of pages reserved for use by programs that run outside the AEMS system (these are the pages used when the AEMS card is operating in "pass mode" as a normal 32K card).

Pressing any key other than A, B or C returns you to the main menu.

Exit to TI Menu

Selecting item D, "Exit to TI Menu", puts the AEMS card in pass mode and exits to the normal TI GPL main menu.

TAILORING THE MAIN MENU

The boot process (program ABOOT or XBOOT) loads the AEMS system programs, AEMSRES1 and AEMSRES2, into memory and builds the main menu. The user items on the main menu are specified via "statements" in file "AEMSSYS".

ABOOT or XBOOT need only be run after a power up of the AEMS memory. Program ASHOE, XSHOE or OSHOE can be run to restart the already loaded AEMS system from the TI main menu. ASHOE is run from E/A Option 5, TIW Option 3 or any compatible loader. XSHOE is an Extend Basic program. OSHOE is an object code file that can be run from E/A Option 3, Mini Memory Option 3, Myarc Floppy Disk Controller CALL IL, or compatible loader.

The AEMSSYS file is a standard text file (V 80) that can be edited using any editor. All statements in the file are coded with a keyword beginning in column one. The keyword is followed by at least one blank and then by any operands. The operands may not contain blanks unless enclosed in single quotes. Operands are separated by commas. A comment may follow the operands, separated from them by at least one blank. The statements are not case sensitive except for strings enclosed in quotes. Comment lines, which begin with an asterisk in column one, may be used freely in the file.

All statements are read into memory before any are processed. This allows changing of disks if necessary for loading programs.

The individual statement format and functions are described below.

RAM Statement

This statement tells the AEMS system how much RAM you have on the AEMS card.

RAM nnnn

where "nnnn" is the number of K bytes of memory you have on the card from 128 to 16384. The RAM statement should be the first statement in the AEMSSYS file, otherwise ABOOT may not be able to execute the LOAD statements.

SW99 Statement

This statement tells the AEMS system that you have a Super AMS card from the Southwest 99ers Group. It enables the mapping of the two pages in low memory. If this statement is not used the AEMS system assumes an AMS card.

SW99

There are no operands on this statement. The SW99 statement should be the second statement in the AEMSSYS file, otherwise ABOOT may not be able to execute the LOAD statements.

MENU Statement

This statement provides a program file name and menu text for an item to be added to the AEMS main menu.

MENU filename,'menu text'

The "filename" is the full device/filename of an E/A Option 5 or AEMS program file. The menu text is limited to 15 characters. When the menu item is selected the program file is loaded into available memory and executed.

LOAD Statement

This statement provides a program file name and menu text for a program that is to be loaded immediately and retained in memory for the session.

The "filename" is the full device/filename of an E/A Option 5 or AEMS program file. The menu text is limited to 15 characters. When the menu item is selected the program in memory is executed immediately.

PATCH Statement

This statement is used to make patches to a program after it has been loaded into memory. Patch statements must immediately follow the LOAD statement that loaded the program. If the

program is an overlay program then patches may be made only to the root segment.

PATCH >addr,patch-data

where "addr" is the hexadecimal address where the patching is to begin, and "patch-data" is the data to be patched into the program. "patch-data" may be specified as a string of characters, enclosed in single quotes if it contains blanks or commas, or a string of hexadecimal digits, preceded by the ">" to indicate hexadecimal notation.

The PAUSE Statement

This statement causes the boot program to suspend processing and await a key press before continuing. This could be used to allow you to change disks in order to access programs to be loaded.

MODIFYING PROGRAMS

This section is intended as an aid for those who want to make their favorite programs resident or compatible with the AEMS system. Programs made resident must conform to the following specifications.

1. Must be E/A Option 5 type or AEMS programs.
2. Must be well behaved. That is, must use defined system resources correctly.
3. In order to be resident a program must be reusable. That is, must not depend upon being reloaded in order to initialize memory areas.

To determine if a program is a well behaved E/A Option 5 program, it can be run using the E/A Cartridge. If when run this way, the program terminates with the Option 5 screen displayed again, then it probably is well behaved. If it exits to the TI title screen it must be modified.

All AEMS programs should be well behaved and should exit directly to the AEMS main menu.

Requirement 3 can best be determined by trying to use the program after making it resident and running it several times.

If you have experience in Assembler Language you may be able to patch a program to make it usable. This is the reason for the PATCH statement. Programs can also be permanently patched using the ZAEMSPAT program.

In order to facilitate exit from standard E/A programs the AEMS loader places some special "exit code" at four places in memory

before loading and execution of the program. The code is placed at:

- >3FF2 - End of low memory,
- >FFF2 - End of high memory,
- >8392 - RAMPAD GPL Stack area,
- >83B2 - RAMPAD GPL Stack area.

Most programs should leave at least one of these areas unmodified so that a branch can be patched into the program to exit to one of these code segments which will exit to AEMS. When programs are executed, GPL R11, the usual return address, is set to >83B2.

OTHER SOFTWARE AVAILABLE

The following is a partial list of programs written especially for AEMS.

AEMS SUBROUTINE LIBRARY and LIBRARY MANAGER

Fairware from RAG Software

The AEMS Subroutine Library contains the Assembler Language routines necessary for programs to interface to the AEMS and the standard TI operating system. The object code of the library is public domain. The AEMS Library Manager allows building and maintenance of LINKER libraries of subroutines.

AEMS OBJECT LINKER

Fairware from RAG Software

The AEMS Object Linker creates program files from Assembler object files. Provides a library search for REFS. Allows building standard E/A Option 5 program files and AEMS overlay program files. Processes COMMON data area definitions. Makes full use of the memory available to build large program files.

AEMS MACRO ASSEMBLER

Fairware from RAG Software

The AEMS Macro Assembler is a full featured 9900 macro assembler. It provides many features beyond the TI assembler and makes full use of the memory available to assemble large programs. Features are: Macro definitions and expansion; compact listing format; symbol cross reference listing; string and floating data types; COMMON data area definition; complex relocation.