ADVANCED VEHICLE TECHNOLOGIES, Inc. AVT inc.

AVT-1850-1

J1850 Development System

User's Manual

1509 Manor View Road
Davidsonville, MD 21035
410-798-4038 (voice) 410-798-4308 (fax)

Revision 'B1'

March, 1996

TABLE OF CONTENTS

1. INTRODUCTION

The AVT-1850-1 Development System consists of the AVT-1850-1 printed circuit board, development environment software known as EOS (Enhanced On-line Software), and example program files. The development system is designed to be installed in and run on a PC-AT or compatible computer. Together, the board and software give the user the ability to design, develop, prototype, test, and demonstrate nearly any aspect of a J1850 VPW multiplex bus.

The Society of Automotive Engineers (SAE) has adopted a specification known as J1850 "Class B Data Communications Network Interface." This specification describes two forms of a multiplex bus structure intended for use in a vehicle. The two forms of this multiplex bus are known as Pulse Width Modulation (PWM) and Variable Pulse Width (VPW). The AVT-1850-1 implements the VPW class of J1850.

1.1 Specifications and Requirements

EOS Software:

Requires:

- PC-AT class machine.
- 386 or greater processor.
- Minimum of 640 Kbytes of memory.
- Minimum of 1 Mbytes of hard disk drive space.
- VGA Monitor.
- Mouse (recommended).
- The software must be run under DOS only. Do not run it in Windows using the MS-DOS window.
- MS-DOS 5.0 or greater.

AVT-1850-1 Board:

- PC-AT compatible.
- 8 bit ISA bus interface.
- Memory mapped, requires 8K (8192) bytes of free memory space.
- Overall size: 4.8 x 13.3 inches.
- Weight: 9 oz.
- +5 volts from host computer, via ISA bus.
- +12 volts from host computer, via ISA bus.
- Power dissipation: < 700 milliwatts.

1.2 Definitions

The following terms are used in this manual.

- Positive logic is used throughout.
- +5 volts indicates a logical '1' (unless otherwise indicated).
- 0 volts or ground indicates a logical '0' (unless otherwise indicated).
- An active high signal is one in which the action occurs as a result of a logical '1'.
- An active low signal is one in which the action occurs as a result of a logical '0'.

2. Installation

The AVT-1850-1 board is a full sized AT compatible board which should fit any full size AT ISA standard 8-bit slot in a PC-AT compatible computer. This is an 8-bit only board and does not require any local bus or other connections to the host computer.

2.1 Software and Hardware Installation

Prior to installing the AVT-1850-1 board in a host computer the user should establish the current configuration of the machine. The board is memory mapped into the host computer and the segment address of the board must be set properly before installation. Therefore, before installing the board, we recommend that the program "INSTALL.EXE" from the installation disk be run. This program assists the user in identifying free memory areas in the host computer. Once those areas are known the segment address of the board can be set and the board physically installed.

2.1.1 Host Memory Configuration

1. Boot the computer normally.
2. Exit from Windows, if necessary. The installation program should be run from the DOS prompt, **not in a Windows shell**.
3. Insert the AVT-1850-1 installation disk in the "A:" (or "B:") floppy disk drive.
4. At the DOS prompt type "A:\INSTALL.EXE" (or "B:\INSTALL.EXE") and press ENTER. Once in the installation program, the user selects "File" from the menu bar. The user then selects the menu item "Start Installation". (Appendix B contains more details on the installation program.)
5. The installation program prompts the user for the disk and directory on the destination drive in which to store the AVT-1850-1 software (the default is "C:\EOS"). In addition, the user is prompted for the number of boards that are to be configured by the installation program. (As many as ten boards are supported by EOS in one PC chassis.)
6. The installation program then searches the host computer memory for any areas not in use and large enough for the AVT-1850-1 board. (Note that the user can request that the installation program list all valid addresses without checking to see if that memory is available. It is recommended that the user <u>not</u> request this option unless the installation of the AVT-1850-1 board has already been accomplished and the user **knows** the correct segment.)
7. When the memory search completes, the installation program displays the memory areas found. The user selects one of the displayed segments. If more then one board had been requested in step 5 then additional prompts for the those segments are displayed one at a time. Each dialog box prompt for the board segment is titled "Board X Segment" where X is the current board number starting with 1 and ending with the number of boards requested. (An interesting feature of the search for segment option is that each segment selected for a board is removed from the list of valid segments for the next board. This feature is not implemented in the second option which requests the that all valid segments be displayed for all boards. This is another reason to exercise caution when requesting that all segments be displayed.)
8. The user should note the memory segment selected for each board in order to be able to set the DIP switches on the AVT-1850-1 boards later in this procedure.
9. The installation program creates the directory specified in step 5. The EOS program, support programs, and sample programs are copied from the installation disk. At the users request,

several “AUTOEXEC.BAT” file modifications occur at this time. (Please refer to Appendix A for the list of the installation files.)

10. The segment address of the board is determined by the settings of DIP switches 1 through 7 of S1 on the AVT-1850-1 board. Using the addresses selected from those found by the installation program set the DIP switches for each board accordingly.

    Only the top two hex digits are used to set the segment address of the board. Switch #1 on S1 is the MSB of the address and the LSB is always ‘0’.

    Example: The segment address is B200 (hex). The DIP switch will be set to B2 (hex) which is (MSB to LSB): 1011 0010. Since the least significant bit is always ‘0’, on the AVT-1850-1 board, the address should be set to 1011 001 (MSB is switch #1). For DIP switch S1 OFF=+5v=logic ‘1’ and ON=gnd.=logic ‘0’.

    Therefore for segment address B200 the DIP switch should be set to:
    OFF - ON - OFF - OFF - ON - ON - OFF (switches #1 through #7).

11. Exit the installation program and shutdown the host computer.

12. Write down the segment address you set on the AVT-1850-1 boards.

13. AVT-1850-1 Board 1 Segment Address: _____
    AVT-1850-1 Board 2 Segment Address: _____
    AVT-1850-1 Board 3 Segment Address: _____
    AVT-1850-1 Board 4 Segment Address: _____
    AVT-1850-1 Board 5 Segment Address: _____
    AVT-1850-1 Board 6 Segment Address: _____
    AVT-1850-1 Board 7 Segment Address: _____
    AVT-1850-1 Board 8 Segment Address: _____
    AVT-1850-1 Board 9 Segment Address: _____
    AVT-1850-1 Board 10 Segment Address: _____

14. DIP switch #8 is factory set to ON and should be left in that position. It controls the ALC line to the on-board microcontroller, and keeps it pulled low. This should not be changed by unqualified persons since it greatly affects the operation of the microcontroller and hence, the board in general. Refer to Section 5.3 for a brief discussion of this control line and references to its operation.

15. DIP switch #9 is factory set to ON. This switch controls the application of a passive resistive load to the BUS pin on the output connector. The load is a single 10.0 K ohm resistor. (This is approximately equivalent to a single node resistance as specified in J1850 to be 10,500 ohms.) This switch may be turned OFF when connected to a multiple node network.

16. DIP switch #10 is not used.

    Note: Modifications made to the “AUTOEXEC.BAT” file do not take effect until the host computer is rebooted.

2.1.2 Installing the AVT-1850-1 Boards

To install the boards you will need to gain access to the expansion slot area of your computer. If necessary, consult the owner's manual for your machine.

1. Perform a normal shutdown of the host computer.
2. Disconnect the power cord to the computer.
3. Using the appropriate hand tools carefully open the computer and gain access to the expansion slot area of the computer.
4. Use caution when handling the AVT-1850-1 board or when touching any internal components of the host computer. All devices are sensitive to ESD (Electro-Static Discharge). Be sure to maintain a ground contact between yourself, the computer, and the AVT-1850-1 to prevent a buildup of static electricity.
5. Select the expansion slot into which you want to install the AVT-1850-1 board.
6. Remove the blank cover associated with the expansion slot selected and keep the screw.
7. Carefully insert the AVT-1850-1 board into the host computer being sure to line up the card edge guides.
8. Insert the card edge connector of the AVT-1850-1 board into the computer's expansion slot connector. Be sure to keep the card edge connector straight and ensure that the card is fully inserted into the mating connector.
9. Re-install the screw saved when the blank cover was removed.
10. Carefully look over the installation to ensure that there are no points of mechanical interference between the AVT-1850-1 board and any other components in the host computer.
11. Repeat steps 5 through 10 until all of the AVT-1850-1 boards have been installed.
12. Re-assemble the host computer housing.
13. Plug back in the host computer.

2.2 Testing the AVT-1850-1 Boards

After completing the configuration and installation of the AVT-1850-1 boards the diagnostics program should be run to be sure the installation was successful.

The diagnostic program is built into the EOS program. To run diagnostics the user first enters EOS by typing in the command "EOS" from the DOS command prompt. The following steps are then performed to ensure the success of the installation.

1. From the EOS menu bar select PROJECT. From the PROJECT pull down menu select the menu item SWITCH BOARDS...
2. EOS displays the Select AVT-1850 Board Dialog box.
3. Select the AVT-1850 board on which to run the diagnostics program. Press the OK button to activate the selection.
4. When the Select AVT-1850 Board Dialog box is dismissed verify that the correct board has been selected by looking at the status bar at the bottom of the EOS screen. The status bar displays the currently active AVT-1850 board.

5. From the EOS menu bar select PROJECT. From the PROJECT pull down menu select the menu item DIAGNOSTICS....
6. EOS displays the AVT-1850 Diagnostics Dialog box.
7. Select the CHECK ALL button on the dialog box. This places an "X" by each diagnostic test in the "Diagnostic Test" group to indicate that all tests should be run.
8. Press the RUN button to start the tests. The complete suite of tests may take several seconds. The working window indicates that the diagnostic tests are running.
9. If any of the tests fail an Error Message Box display gives the test and the reason for the failure. The user should note this message for later reference. In addition, the "Results" group entry for the test that failed is marked with a FAILED.
10. When a test succeeds, the "Results" group entry for that test is marked with as "PASSED". If all the "Results" entries are marked "PASSED" then the installation of the selected AVT-1850 board completed successfully.
11. Repeat steps 1 through 10 for each board installed.

Refer to Section 7.2 for the telephone number for Advanced Vehicle Technologies Technical Assistance.

3. Operation Overview

The AVT-1850-1 Development System consists of software, in the form of the development environment known as EOS, and hardware, the AVT-1850-1 board. When installed in a host computer these two work together to provide the user with a powerful tool for the development and analysis of SAE standard J1850 VPW multiplex networks and nodes.

The AVT-1850-1 board utilizes an embedded microcontroller known as the HIP7030A0, manufactured by Harris Semiconductor. The HIP7030A0 is a sister to the HIP7030A2 device and both devices are based on the 68HC05 8 bit microprocessor. The A2 device utilizes mask ROM for high volume low cost applications and has no external address, data, or control signals. The A0 device utilizes internal RAM and external memory to emulate the A2 device. Both devices possess identical capabilities including functions unique to the J1850 VPW bus application as well as several multi-purpose interfaces.

The designer/developer has access to and control over the various hardware interfaces of the HIP7030A0 microcontroller on the AVT-1850-1. All of these hardware interfaces are accessible to the user via a 25 pin connector. Thus, a prototype application for the HIP7030A2 can be designed, connected, and tested using the AVT-1850-1.

In a similar fashion, the user has complete control over the design and implementation of the software destined for the HIP7030 microcontroller. The development environment, EOS, provides a means to write, assemble, download, test, examine, and evaluate the software under development. Software can be loaded, tested, changed, and re-loaded repetitively without the need to burn PROM's or change devices.

With the proper microcontroller software and attached hardware, the board can be programmed to function as a node of any type. The user is free to design any code and attach any hardware and thus define the node function. The AVT-1850-1 Development System offers unique flexibility, in both hardware and software, to integrate, construct, test, and demonstrate an application for the HIP7030A2 device before having to commit to mask ROM.

Several sample assembly language programs are supplied as part of the installation of the AVT-1850-1 Development System. These programs can be found in the "\SAMPLES" sub-directory of the installation directory. The source code has a file spec of ".S" and the assembler output file spec is ".H". It is the ".H" file that is downloaded to the on-board memory for the HIP7030A0 device. A header at the beginning of each source (".S") file describes the program.

The AVT-1850-1 board can be directly connected to any SAE standard J1850 VPW bus. The bus and ground connection are available via the 25 pin connector mounted on the AVT-1850-1 board. Refer to Section 7.4 for a complete pinout listing.

4. Introduction to EOS

Included with the AVT-1850-1 Development System is a development environment known as EOS, for Enhanced Operating System. The EOS software is the major component of the software package included with the AVT-1850-1. EOS is a DOS program that can be run from the DOS prompt.

NOTE: EOS must be run under DOS, not under Windows in a DOS box.

EOS was designed to assist the user while developing software on the AVT-1850-1. Through EOS, the designer/developer can control and monitor all aspects of the AVT-1850-1 hardware as well as edit, assemble, and download code onto the board for testing.

EOS is a windowing type of environment offering pull down menus and windows.

EOS supports up to ten boards in a single PC chassis. While all the boards can be tasked through a single EOS session, only one board has the focus of EOS at a time. The board that currently has focus is said to be the active board in the EOS environment. (While a board is active, the other boards still retain any information downloaded when they were active. No on-board information is lost by changing focus.) The status line at the bottem of the EOS screen indicates which board is currently active.

4.1 Using EOS

It is recommended that EOS be used with a mouse. However, a mouse is not required. When using a mouse the usual point and click methods are all that is required. Just move the mouse to move the highlight onto the selection and click the left mouse button.

EOS may also be used without a mouse. To make a menu bar selection, simultaneously press the ALT key and the highlighted letter key of the selection to be made. In pull down menus only the letter key corresponding to the highlighted choice need be pressed.

Choices contain a highlighted letter only if that choice is available for the active window.

Once in a window, the tab key may be used to move from one field to another. Likewise, simultaneously pressing the ALT key and the letter key corresponding the highlighted letter of the desired field will move the highlight to that field. The arrow keys allow movement within a field.

Some menu choices have related 'speed keys.' If a function has a 'speed key' it is displayed in the menu to the right of the menu selection.

Some of the more commonly used functions have 'speed keys' assigned. These 'speed keys' are simple, one-step, methods of accomplishing a function or opening a desired window.

4.2 Starting EOS

During the software installation several directories and sub-directories were created and files were loaded into them. EOS was loaded into the "EOS" directory (unless otherwise specified during the installation). You must move to the "EOS" directory or have a path branch to it (set in the AUTOEXEC.BAT file upon boot) prior to attempting to start EOS. If one of these conditions is not met, when trying to start EOS you will get a DOS error message: "Bad command or file name.".

Proper operation of EOS requires several environment variables be declared and properly set. These variable are normally declared and set in the AUTOEXEC.BAT file using the "SET" command. The following statements should be in your AUTOEXEC.BAT file.

```
SET AVT1850SEG=0xB200                (or the actual board segment address)
SET EOSHOME=C:\EOS                   (or the installation directory)
SET EOSASSEMBLER=C:\EOS\HASM.EXE
SET EOSBOOTPGM=C:\EOS\INIT.H
```

NOTE: The installation program will update the "AUTOEXEC.BAT" file with the necessary environment variables and to the Path variable.

The above example shows the environment variables for a one board setup. If multiple AVT-1850-1 boards are installed in the system then each of those boards must have a corresponding environment variable. The first board in the system is defined by the variable AVT1850SEG, the second board by AVT1850SEG1, the third by AVT1850SEG2, and so on. Up to ten boards can be defined in a single PC chassis. The tenth board has the environment variable AVT1850SEG9. (Note: no gaps can appear in the environment variable assignment. For example, if board three is defined as AVT1850SEG2 and the next board is AVT1850SEG4, then only the first three boards are defined. It is not necessary to define the boards in order in the environment space, only that the end result has no gaps.)

At the DOS prompt type: "EOS" and press Enter. If all of the environment variables are properly declared and set, then EOS will start and the main window with menu bar will appear. The first AVT-1850-1 board, defined by the environment variable AVT1850SEG, becomes active as will be noted on the status bar at the bottem of the screen. At this point EOS is ready to be used.

If no segments are defined in the environment space an error is displayed on EOS startup. After the operator acknowledges the error the EOS program is shutdown.

4.3 Window and Menu Descriptions

In the following sections a brief description of each of the menu bar items and sub-items are described.

4.3.1 FILE

Selecting FILE from the main menu bar opens up the pull down menu and offers typical file manipulation choices. The user may select among: Open, New, Save, Save as, Change dir., DOS shell, and Exit. Selection of a menu choice opens the corresponding window.

A brief description of each of the menu choices is contained in the following sections.

4.3.1.1 Open

The OPEN FILE window is opened. At the top of the window is the Name line. Fill in the name of the file to be opened or move to the Files list and select the file to be opened. In the Files list are also listed all of the sub-directories as well as the DOS "\.." operator, if applicable.

The bottom of the screen is displayed the current directory as well as details about the highlighted file.

The selected file is opened for text editing.

4.3.1.2 New

This selection opens a new, untitled, editing window.

4.3.1.3 Save

Saves, or writes, the contents of the current window to the corresponding file.

4.3.1.4 Save as

Allows the user to specify the name of the file in which to save contents of the current window.

4.3.1.5 Change dir.

In this window the user can select the working directory to be used in this session. The specified directory is used when the Open file window is displayed. To set the working directory, select the directory or sub-directory, select CHDIR and the working directory will be displayed. Selecting OK will close the window and the working directory will be set.

4.3.1.6 DOS shell

This menu choice suspends the EOS program and enters a DOS shell. To return to EOS, simply type EXIT and press enter. EOS execution will resume.

4.3.1.7 Exit

This terminates the EOS program and returns to the DOS prompt environment.

4.3.2 EDIT

Selecting EDIT from the main menu bar opens up the pull down menu and offers typical text editing choices. The user may select among: Undo, Cut, Copy, Paste, Show clipboard, and Clear.

A brief description of each of the menu choices is contained in the following sections.

4.3.2.1 Undo

Undoes the last editing action.

4.3.2.2 Cut

Cuts the selection and places it in the clipboard.

4.3.2.3 Copy

Copies the selection into the clipboard. Leaves the selection unaffected.

4.3.2.4 Paste

Pastes or inserts the contents of the clipboard into the active window at the location of the cursor.

4.3.2.5 Show clipboard

Opens the clipboard window and displays the contents of the clipboard.

4.3.2.6 Clear

Deletes the selected, or highlighted text.

4.3.3 SEARCH

Selecting SEARCH from the main menu bar opens up the pull down menu and offers typical text search and replace editing choices. The user may select among: Find, Replace, and Search again.

A brief description of each of the menu choices is contained in the following sections.

4.3.3.1 Find

This selection opens the FIND window. The text string to be searched for is entered in the 'Text to find' line. Selecting the down arrow displays a history of searches. Additionally, case sensitivity and whole words only search criteria can be selected.

4.3.3.2 Replace

This selection opens the REPLACE window. The user fills in the 'Text to find' and the 'New text' lines. Down arrows display histories for the respective lines. Search and replace criteria that can be selected are: Case sensitivity, Whole words only, Prompt on replace, Replace all.

4.3.3.3 Search again

This selection repeats the last search specified using the Find window.

4.3.4 PROJECT

Selecting PROJECT from the main menu bar opens up the pull down menu and offers the following choices: Download, Assemble, and Status.

A brief description of each of the menu choices is contained in the following sections.

4.3.4.1 Download

This selection opens the DOWNLOAD FILE window. Selecting the appropriate file and then selecting OK results in the downloading of that file into the memory of the AVT-1850-1. This function is used to download a file containing the machine code output from the assembler. The file

choices default to "*.H" files. These are the normal output files from the assembler and are in Motorola 'S' format.

Once a file has been selected, OK is selected, and the file is downloaded onto the AVT-1850-1 a confirmation window is displayed showing what file was downloaded and if the download was successful.

4.3.4.2 Assemble

This selection opens the ASSEMBLE window. The name of the file to assemble is displayed and is the current active window file. To obtain a listing file (file spec ".LST") check the Listing File selection in the ASSEMBLE window. Selecting OK invokes the assembler as specified by the EOSASSEMBLER environment variable. As a default, this assembler is the Harris Semiconductor 68HC05 assembler.

When the assembler is invoked a new DOS window is opened to display the output of the assembler as the file is assembled. When assembly has been completed the message: "Press any key to continue" is displayed. Pressing a key closes the assembly window and opens the ASSEMBLY RESULTS window. This window contains the output of the assembler, including any error messages, if there are any. The ASSEMBLY RESULTS window can be closed to return to the original active window.

4.3.4.3 Status

Selecting STATUS opens the "EOS STATUS" window. In this window is listed the following:

Last File Downloaded,
Status Register (in hex),
Board Segment Address,
EOS Home Directory,
EOS Bootstrap Program,
EOS Assembler.

4.3.4.4 Switch Boards

This purpose of the SWITCH BOARDS selection is to allow the operator to switch focus between multiple installed AVT-1850-1 boards. Invoking this item displays the "Select AVT-1850 Boards" dialog box. This dialog lists all the boards in the system using the corresponding environment variable name as defined in the AUTOEXEC.BAT file. To switch focus to a particular board the operator highlights the board of interest. When the OK button is pressed the highlighted board receives the focus and the dialog box is dismissed; selecting CANCEL dismisses the box without changing the current focus.

It should be noted that changing focus does not dismiss any active displays in the EOS environment. These displays are now active for the board currently in focus not the board that had focus when the displays were invoked. For example, if the network monitor was running and displaying output, switching focus would not dismiss the network monitor dialog box. Furthermore, since the board in focus did not start a network monitor program no display updates would now occur. It should be noted that the previous board is still running the network monitor program but since EOS is no longer focused on that board it can no longer read the network data supplied from it. The data could overflow and be lost before the operator has the opportunity to switch back to the board previously in focus.

The network monitor dialog box represents a special case in task switching since it is one of the few dialogs that download a program and expects to read data from the active program. It is covered in length in this section because it illustrates an extreme case of task switching. For most cases the operator does not need to be concerned about switching and losing data. It is important to remember, however, that the EOS environment interfaces to one or more dynamic boards which may be reading and writing data in real-time. The operator should plan the order of downloads and task switching in order to ensure the optimal information flow with a minimum of data loss.

4.3.4.5 Diagnostics

This selection opens the "AVT-1850 DIAGNOSTICS" window. In this window the various diagnostic routines can be selected and run. The results of the selected tests are also displayed. Refer to Section 2.2 for more information on running the board level diagnostics.

4.3.5 MONITOR

Selecting MONITOR from the main menu bar opens up the pull down menu and offers the following choices: Status Registers, FIFO In, FIFO Out, and RAM.

A brief description of each of the menu choices is contained in the following sections.

4.3.5.1 Status Registers

This selection opens the STATUS REGISTERS window. This window is divided into four sections.

The 'Status Register' section is a display only section. In it the various bits of the read only hardware control register are displayed.

In the 'Reset FIFO's' section, either or both FIFO's can be selected for reset. After selecting the FIFO or FIFO's for reset, 'Update' must then be selected. The selected FIFO's are then reset and the checks in the 'Reset FIFO's' section are cleared.

In the 'Write Only Values' section the 'Loopback' and/or the 'FIFO 1 Overflow' bits of the control register can be manually or cleared. After making the appropriate selection, the 'Update' button must be clicked to affect the changes. The 'Loopback' control bit controls the loopback function of the HIP7020 J1805 VPW bus transceiver device.

The 'AVT-1850 Control' section allows manual control of the status of the HIP7030A0 microcontroller on the AVT-1850 board. Selecting 'Start' and then selecting 'Update' will result in the release of the RESET line of the HIP7030A0 microcontroller. At that point the microcontroller will fetch the reset vector from the top of memory and commence execution.

While the window is open the status of the control register is updated periodically. However, by selecting the "Update" button the display is updated immediately.

Once a selection has been made, select Update to write the new value to the control register.

4.3.5.2 FIFO In

This selection opens the FIFO IN window. This window is used to read bytes from FIFO #2 and display them. When the user selects Get Byte a single byte is read from FIFO #2 and displayed, in hex format, in the Bytes Read display area. If the FIFO is empty, a message is displayed informing the user.

4.3.5.3 FIFO Out

This selection opens the FIFO OUT window. The user enters bytes, in hex format, on the Bytes to Send line. By selecting Send, the entered bytes are then written to FIFO #1. The Clear button is used to clear the Bytes to Send line (not the FIFO). Repetitively selecting Send writes the information to the FIFO multiple times.

4.3.5.4 RAM

This selection opens the RAM window. In this window the user selects the memory space on the AVT-1850-1 board to be examined. By selecting the area of memory to be examined (refer to Section 5.2 for a discussion of memory organization) and then selecting Update, the contents of that memory area is displayed in the Dump window, in hex format. The entire memory selected can be scrolled through for examination.

4.3.5.5 Network Monitor

This selection opens the Network Monitor dialog box which is responsible for reporting network traffic to the operator. In order to start the network monitor function, the operator presses the START/UPDATE button. If the network monitor was not previously running, this selection downloads the network monitor program to the AVT-1850-1 board, starts the downloaded network program, and reads the network traffic supplied by the AVT-1850-1. Network data is displayed in the list box named "Network Activity". If the log function was selected (see below) then the data is also logged to a disk file. Selecting the START/UPDATE button when the network monitor function is already active modifies the current monitor session to reflect operator changes to the checkbox buttons. This is covered in more detail later in this section.

The STOP button stops the execution of the network monitor program. The program on the AVT-1850-1 board is sent a stop message but no open files are closed. If the operator was to press the START/UPDATE button again, the log data would continue to the same file.

The CLOSE button dismisses the Network Monitor program. The network monitor program in the AVT-1850-1 is sent a stop message and the open log file (if any) is closed.

Two checkboxes are available in the Network Monitor dialog box. The first, titled "Timetag Network Monitor Data" permits the operator to time tag any data displayed or written to the log file. This time tag has a granularity of one second so several messages may have the same time; however, the messages are always displayed in the order received.

The second checkbox is titled "Log Network Data to Disk". This option requests that all network data be written to a disk file. The file is located in the EOS working directory and has the filename MMDDHHMM.LOG where MM represents the month, DD the day, HH the hour, and MM the minute that the log file was created. This file is created when the START/UPDATE button is pressed and remains open until the network monitor dialog box is dismissed or until the operator deselects the "Log Network Data to Disk" checkbox and presses the START/UPDATE button.

It should be noted that the "Network Activity" list box only displays data once every second or so; if the network activity results in higher message traffic then the excess data is not displayed. This is necessary in order to ensure that EOS is not bogged down by terminal I/O. If the operator has checked the "Log Network Data to Disk", however, all network traffic is saved to the log file**. No data is lost in writing network data to disk.**

4.3.6 WINDOWS

Selecting WINDOWS from the main menu bar opens up the pull down menu and offers typical window management choices. The user may select among: Size/move, Zoom, Tile, Cascade, Next, Previous, and Close.

A brief description of each of the menu choices is contained in the following sections.

4.3.6.1 Size/move

With this selection the user can move the location of the active window using the keyboard arrow keys and change the size of the window using SHIFT + the keyboard arrow keys.

4.3.6.2 Zoom

With this selection the size of the window toggles between a full screen window and a smaller window, as set using the Size/move function.

4.3.6.3 Tile

If multiple windows are open, this selection will automatically size the windows, vertically, such that all of the windows are visible on the screen at once.

4.3.6.4 Cascade

If multiple windows are open, this selection will automatically move the windows such that the upper left corners, including the window title bar, of each of the windows is displayed. The user can then use the mouse to click on a title bar to select that window.

4.3.6.5 Next

This selection changes the active window to the next open window.

4.3.6.6 Previous

This selection changes the active window to the previous open window.

4.3.6.7 Close

This selection closes the current active window. If changes have been made, but not saved, the user is informed prior to closing the window.

4.3.7 HELP

Selecting HELP from the main menu bar opens up the pull down menu and offers the About choice.

4.3.7.1 About

This selection opens the ABOUT EOS information window. Displayed in this window is the name of the program, the version number, the date of the version, and copyright information.

5. Hardware Description

This section contains a discussion of the AVT-1850-1 board. The architecture of the board is based on the Harris HIP7030A2 device, which is a mask ROM microcontroller designed for embedded applications. However, a designer needs to be able develop and test code and hardware prior to committing to a production run of mask ROM devices. It is this development function that the AVT-1850-1 Development System was designed to fulfill.

The AVT-1850-1 uses the HIP7030A0 which is an emulator version of the A2 device. The board was designed around the A0 device in such a way as to mimic, as closely as possible, the operation of the A2 device. One intent of the AVT-1850-1 board was to allow the user to develop hardware and software on the AVT-1850-1 and then be able to transition directly to the mask ROM A2 device.

To improve the flexibility and usefulness of the AVT-1850-1, additional memory and control functions were added to the board that have no direct equivalency with the A2 device. These additional features were mapped into an unused/undefined region of the A2 device. By utilizing these additional features, the AVT-1850-1 can be used to implement functions other than node prototype development, such as a network monitor.

5.1 Board Segment Address

To the host computer, the board appears as an 8k byte memory space at a segment address determined by the setting of the first 7 DIP switches on S1. In theory, the AVT-1850-1 board can reside at discrete segment addresses in the range from 0x0000h to 0xFE00h. Since the board occupies 8k bytes (8192) of space the available segment addresses increment by 0x200h.

Due to restrictions in the architecture of PC's the board should not be set to a segment address below 0xA000h, as this is the 640k byte boundary sacred to DOS. Likewise the board segment address should not be set above 0xFE00h, as a higher address would put the board memory above the 1 megabyte memory boundary.

There are a number of other restrictions on where the board can reside in the host PC memory space due to the location of video memory and the presence of other expansion cards. These restrictions are highly dependent on the particular machine into which the board is currently being installed.

The installation program will search for available memory space based on these restrictions and other criteria. The program searches for available memory by looking for unused memory space which is often indicated by reading a value of 0xFFh.

NOTE: Throughout all discussions of board memory addresses, from the host PC point of view, the board segment address is not specified and is assumed to be set properly.

5.1.1 PC Memory Refresher

For PC and compatible computers memory is byte organized (8 bits wide). The number of address lines is dependent on the processor being used as well as the manufacturer/designer of the computer motherboard. Expansion boards, such as the AVT-1850-1, which use the ISA expansion slots only have access to address lines A0 through A19. Therefore, expansion boards must lie in address space from 0x00h through 0xFFFFFh.

From the point of view of the processor a physical address is accessed by specifying the segment address and then the offset address. The segment address is, essentially, shifted one byte to

the left and then added to the offset address to obtain the effective or physical address. This is the address that is put out on the processor's address lines.

As stated earlier, the first 640K of memory in a PC is sacred to DOS and, for the purposes here, should not be used or modified. Also, due to the number of address lines available at an ISA expansion slot connector the addresses available must be below 1 Mbyte. Therefore, for the AVT-1850-1 the segment address must be between 0xA000h and 0xFE00h.
[0xA000h segment address corresponds to an effective address of: 0xA0000h.
Where 0xA0000h = 655360 = 640*1024 = 640 Kbytes.]

The AVT-1850-1 occupies 8 Kbytes of memory space (0x1FFFh). Therefore, the board was designed to only lie at even 0x2000h boundaries in address space (which corresponds to 0x200h increments in segment address). For example the AVT-1850-1 board segment address can be set to 0xA000h, 0xA200h, 0xA400h, ..., 0xB200h, 0xB400h, ..., 0xFE00h.

5.2 Memory

The AVT-1850-1 board uses several different types of memory organized in such a way that to the HIP7030A0 device it 'looks' like the memory space in an HIP7030A2 device.

To the HIP7030A0 device the board memory space is a linearly organized 8k (8192) bytes. The memory is directly accessed by the HIP7030A0 device via its 13 address lines (A0 through A12). The HIP7030A0 device does not use segmented memory. The address range of the available memory, from the HIP7030A0 device point of view, is from 0x0000h through 0x1FFFh.

To the host PC the board is accessed by specifying the segment address and then the offset address. The valid range of offset addresses for locations on the AVT-1850-1 board are 0x0000h through 0x1FFFh and correspond directly with the address space as 'seen' by the HIP7030A0 microcontroller.

Refer to Section 5.2.8 for a complete map of the memory of the AVT-1850-1 board. The memory map is specified for both the HIP7030A0 device and the host PC. The memory map also indicates what areas are read only and what areas are read/write, with respect from the point of view of the HIP7030A0 and the host PC.

5.2.1 HIP7030A0 ROM Space

RAM #3

Memory space in the range of 0x100h through 0x1BFFh exists as ROM to the HIP7030A0 device. Into this memory space is loaded the machine code for the microcontroller, by EOS, from the host PC. There is no affect to the contents of memory in this space should the HIP7030A0 device attempt to perform a write operation here.

RAM #1

Memory space in the range of 0x1E00h through 0x1FFFh exists as ROM to the HIP7030A0 device. At the top of this memory space is typically loaded the microcontroller interrupt vectors. There is no affect to the contents of memory in this space should the HIP7030A0 device attempt to perform a write operation here.

5.2.2 Host PC ROM Space

RAM #2

Memory space in the range of 0x1C00h through 0x1DFBh exists as ROM to the host PC. There is no affect to the contents of memory in this space should the host PC attempt to perform a write operation here.

5.2.3 HIP7030A0 RAM Space

RAM #4

Memory space in the range of 0x00h through 0xFFh contains both ROM and RAM regions, to the HIP7030A0. Physically, this memory exists internal to the HIP7030A0 device and cannot be 'seen' by the host PC.

The various control registers for the HIP7030A0 are located in this memory space as is RAM. The RAM space in this area is used by the stack and is also available for general purpose use by the developer (scratch pad area).

Refer to Section 5.2.8 for additional details on the organization of this memory area in the HIP7030A0 device. Also refer to the Harris Corp. Technical literature on the HIP7030A2 device for information on the organization of memory in this area and details on the various control register functions.

RAM #2

Memory space in the range of 0x1C00h through 0x1DFBh exists as RAM space to the HIP7030A0 device. This RAM space is available for unrestricted read/write use by the developer. It should be noted that this space does not exists in the HIP7030A2 device.

5.2.4 Host PC RAM Space

RAM #3

Memory space in the range of 0x100h through 0x1BFFh exists as RAM space to the host PC. Normally EOS is used to load the HIP7030A0's machine code into this area. Individual memory locations in this area can be modified and monitored, by the user, using EOS. The developer is free to use this space in any way.

RAM #1

Memory space in the range of 0x1E00h through 0x1FFFh exists as RAM space to the host PC. Normally EOS is used to load the HIP7030A0's interrupt vectors into the top of this memory space. EOS can also be used to modify specific locations as well as read back the contents of memory in this area.

5.2.5 Host PC NVRAM Space

RAM #5

Memory space in the range of 0x00h through 0xFFh, to the host PC, is occupied by a battery backed RAM device. The write enable line of this device is normally inhibited. Thus, this space normally exists as ROM. In it may be stored information to be used by the factory or by other test and diagnostic programs. Additionally, this device may store information used by the host PC BIOS during POST (power on self-test) operations.

This memory space cannot be 'seen' by the HIP7030A0 device.

It is not recommended that the contents of this device be altered. Should the contents of this device be changed, Advanced Vehicle Technologies, Inc. cannot be held responsible for the consequences, which cannot be predicted.

5.2.6 FIFO

A FIFO (a First In First Out) device is, by its very nature, a read only device when viewed from the input side and a write only device when viewed from the output side. The AVT-1850-1 board uses two FIFO's for bi-directional message passing between the host PC and the HIP7030A0 device.

One FIFO is oriented to allow message passing from the host PC to the HIP7030A0 device while a second FIFO is oriented to allow communications from the HIP7030A0 device to the host PC. Each FIFO occupies only one byte of address space, yet allows the storage of 2k bytes of information.

FIFO #1 resides at address 0x1DFFh and is a write only device for the host PC. Likewise, FIFO #1 is a read-only device for the HIP7030A0 device.

FIFO #2 resides at address 0x1DFEh and is a write only device for the HIP7030A0 device. Similarly, FIFO #2 is a read only device for the host PC.

The FIFO's are identical units, 8 bits wide, and 2k (2048) bytes deep. The FIFO's used are actually SRAM devices with on-chip address decoders for read and write pointers and do not suffer from 'ripple' delay like other designs. Each FIFO also includes Full and Empty flag outputs as well as a Reset input. The Reset input to the FIFO is asynchronous.

If a FIFO is written to while the Full flag is active, the data is lost and the data stored in the FIFO is unaffected. If a read operation is attempted while a FIFO Empty flag is active, the data bus is not driven but is, instead, left in a high impedance state. (Generally, in a PC, a read operation conducted on a data bus that is not driven will usually result in a read of 0xFFh.)

The FIFO Full and Empty flag outputs as well as the Reset input are all active low signals. Therefore, a logic '0' written to the Reset causes the FIFO to asynchronously enter the reset state. This causes the Empty flag to become active, a '0' is output, and the Full flag to go inactive, a '1' is output. The FIFO will not exit the reset state until the Reset line is returned to an inactive state, by writing a '1' to the Reset bit.

The developer has access, from either the host PC or the HIP7030A0 device, to the Reset inputs of both FIFO devices. To the host PC, in Control Register #2, bit #3 (0x8h) is the Reset signal to FIFO #1 and bit #4 (0x10h) is the Reset signal to FIFO #2. To the HIP-7030A0 device, in Control Register #1, bit #3 (0x8h) is the Reset signal to FIFO #1 and bit #4 (0x10h) is the Reset signal to FIFO #2.

Both the host PC and the HIP7030A0 device must release the Reset signal to a FIFO before the FIFO can be released from a reset condition. For example, both the host PC and the HIP7030A0 device have to write a logical '1' to bit #3 of Control Register #2 and #1, respectively, to release FIFO #1 from a Reset condition.

The Full and Empty flag outputs from each FIFO are routed to the two read only Control Registers. Control Register #1 is a read only location to the host PC and, likewise, Control Register #2 is a read only location to the HIP7030A0 device. Both the host PC and the HIP7030A0 device can read the status of the two FIFO's independently.

In the Control Registers, bit #1 and bit #2 are the Full and Empty flags, respectively, for FIFO #1. Likewise, bit #5 and bit #6 are the Empty and Full flags, respectively, for FIFO #2.

It is left to the developer to ensure that when a FIFO is used, that the status of the device to be used is checked prior to initiating an operation with that device; read or write.

5.2.7 Control Registers

There are two locations in memory that are occupied by special registers designed, primarily, to control board hardware functions. Both the host PC and the HIP7030A0 device can access these two registers via a normal memory operation. However, to the host PC one location, at address 0x1DFCh and occupied by Control Register #2, is a write-only location and that same location is a read-only location to the HIP7030A0 device. Similarly, Register #1, at address 0x1DFDh, is a read-only location to the host PC and a write-only location to the HIP7030A0 device.

Most of the bits in these two registers are dedicated to specific hardware functions. There are some bits that do not have hardware definitions and are passed through from the write side to the read side. Additionally, there are several bits that are either not used or have a fixed value. The developer should refer to the following sections or to the memory map for additional details on these bit definitions. The user should ensure that they understand the active levels of these bits, as some are active low and others are active high.

Register #2, bit #0, is of particular interest. When the host PC writes a logic '0' to this location the Reset pin of the HIP7030A0 device is held active (low) and thus the HIP7030A0 is held in a Reset state (i.e. it is halted). Conversely, when the host PC writes a logic '1' to this location the HIP7030A0 device is released from the Reset state and allowed to run. This particular signal is configured to always initiate to the Reset state (logic '0') upon board power-up. This prevents the

HIP7030A0 from running on power-up and attempting to execute the random configuration of the board RAM / ROM as if it were code.

5.2.7.1 Detailed Description of Control Register #1:

This register physically exists at address 0x1DFDh. Only the HIP device can write to this register and only the PC can read this register. Refer to the Memory Map in Section 5.2.8. The individual bits of these two registers are described here. Bits indicated as 'Not connected' are functional "Don't Cares."

HIP write only

b0: Not connected.
b1: Not connected.
b2: Not connected.
b3: FIFO #1 Reset. Active low. A logic '0' forces FIFO #1 into reset.
b4: FIFO #2 Reset. Active low. A logic '0' forces FIFO #2 into reset
b5: This output is not connected.
b6: This output is not connected.
b7: This bit is connected directly to the read side bit #7.
This bit is functionally defined to be FIFO #2 Overflow. Used to signal that a write operation was desired to FIFO #2, but the FIFO was found to be full. Defined to be active low (a logic '0' indicates that a FIFO overflow occurred).
[Functional definition of this bit is controlled by the user.]

PC read only

b0: HIP reset bit, read from control register #2, b0.
b1: FIFO #1 Full flag. A logic '0' indicates that FIFO #1 is full.
b2: FIFO #1 Empty flag. A logic '0' indicates that FIFO #1 is empty.
b3: FIFO #1 Reset. A logic '0' indicates that FIFO #1 is in reset.
b4: FIFO #2 Reset. A logic '0' indicates that FIFO #2 is in reset.
b5: FIFO #2 Empty flag. A logic '0' indicates that FIFO #2 is empty.
b6: FIFO #2 Full flag. A logic '0' indicates that FIFO #2 is full.
b7: This bit is connected directly to the write side bit #7.
This bit is functionally defined to be FIFO #2 Overflow. Used to signal that a write operation was desired to FIFO #2, but the FIFO was found to be full. Defined to be active low (a logic '0' indicates that a FIFO overflow occurred).
[Functional definition of this bit is controlled by the user.]

5.2.7.2 Detailed Description of Control Register #2:

This register physically exists at address 0x1DFCh. Only the PC can write to this register and only the HIP can read this register. Refer to the Memory Map in Section 5.2.8. The individual bits of these two registers are described here.

PC write only

b0: HIP7030A0 Reset. A logic '0' holds the HIP device in a reset state.
b1: Not connected.
b2: Not connected.
b3: FIFO #1 Reset. Active low. A logic '0' forces FIFO #1 into reset.
b4: FIFO #2 Reset. Active low. A logic '0' forces FIFO #2 into reset
b5: Loopback. A logic '0' enables the bus driver internal loopback. When loopback is enabled, the board is effectively disconnected from the bus yet transmitted symbols are received and routed back to the HIP7030A0's input.
b6: ROM write enable. A logic '1' allows write operations to the NVRAM device, also known as RAM #5. A logic '0' is normally written to this location.
b7: This bit is connected directly to the read side bit #7.
This bit is functionally defined to be FIFO #1 Overflow. Used to signal that a write operation was desired to FIFO #1, but the FIFO was found to be full. Defined to be active low (a logic '0' indicates that a FIFO overflow occurred).
[Functional definition of this bit is controlled by the user.]

HIP read only

b0: Always a logic '0'.
b1: FIFO #1 Full flag. A logic '0' indicates that FIFO #1 is full.
b2: FIFO #1 Empty flag. A logic '0' indicates that FIFO #1 is empty.
b3: FIFO #1 Reset. A logic '0' indicates that FIFO #1 is in reset.
b4: FIFO #2 Reset. A logic '0' indicates that FIFO #2 is in reset.
b5: FIFO #2 Empty flag. A logic '0' indicates that FIFO #2 is empty.
b6: FIFO #2 Full flag. A logic '0' indicates that FIFO #2 is full.
b7: This bit is connected directly to the write side bit #7.
This bit is functionally defined to be FIFO #1 Overflow. Used to signal that a write operation was desired to FIFO #1, but the FIFO was found to be full. Defined to be active low (a '0' indicates that a FIFO overflow occurred).
[Functional definition of this bit is controlled by the user.]

5.2.8 Memory Map

The following pages list all of the memory locations accessible by both the host PC and the HIP7030A0 device. Each of the memory areas and special locations are listed individually. Also indicated is the permitted operations (read-only, write-only, or read-write) of each of these areas with respect to both the host PC and the HIP7030A0 device.

Remember that to the host PC the address of each board memory location is relative to the board's segment address (factory default setting to 0xB200h).

Board Memory Map

Description		Op's	PC Address (+ Segment)	7030 Address	Op's		Description
Dual Port RAM #1 (512 bytes)		R/W	001FFFh 001E00h	01FFFh 01E00h	R		Dual Port RAM #1 (512 bytes)
Dual Port FIFO #1 (2048 bytes)		W	001DFFh	01DFFh	R		Dual Port FIFO #1 (2048 bytes)
Dual Port FIFO #2 (2048 bytes)		R	001DFEh	01DFEh	W		Dual Port FIFO #2 (2048 bytes)
H/W Control Register Latch #1			001DFDh	01DFDh			H/W Control Register Latch #1
! FIFO #2 Overflow	80h	R				80h	! FIFO #2 Overflow
! FIFO #2 Full Flag:	40h	R				40h	n/c
! FIFO #2 Empty Flag:	20h	R				20h	n/c
! FIFO #2 Reset:	10h	R			W	10h	! FIFO #2 Reset
! FIFO #1 Reset:	08h	R			W	08h	! FIFO #1 Reset
! FIFO #1 Empty Flag:	04h	R			W	04h	n/c
! FIFO #1 Full Flag:	02h	R			W	02h	n/c
! 7030 Reset:	01h	R				01h	n/c
H/W Control Register Latch #2			001DFCh	01DFCh			H/W Control Register Latch #2
! FIFO #1 Overflow	80h	W			R	80h	! FIFO #1 Overflow
ROM write enable bit:	40h	W			R	40h	! FIFO #2 Full Flag
! Loop Back:	20h	W			R	20h	! FIFO #2 Empty Flag
! FIFO #2 Reset:	10h	W			R	10h	! FIFO #2 Reset
! FIFO #1 Reset:	08h	W			R	08h	! FIFO #1 Reset
n/c	04h				R	04h	! FIFO #1 Empty Flag
n/c	02h				R	02h	! FIFO #1 Full Flag
! 7030 Reset:	01h	W				01h	always low

Description	Op's	PC Address (+ Segment)	7030 Address	Op's	Description
Dual Port RAM #2 (508 bytes)	R	001DFBh 001C00h	01DFBh 01C00h	R/W	Dual Port RAM #2 (508 bytes)
Dual Port RAM #3 (6912 bytes)	R/W	001BFFh 000100h	01BFFh 00100h	R	Dual Port RAM #3 (6912 bytes)
			000FFh 00000h	R/W	Internal RAM #4 (Internal Page 0) (256 bytes)
ROM (Battery backed RAM) #5 (256 bytes)	R/W	0000FFh 000000h			

R/W	=	Read and Write
R	=	Read only
W	=	Write only
n/a	=	no access
n/c	=	no connection
x	=	don't care
!	=	Boolean NOT

Base Segment Address, switch selected with Y = 00h to 0Dh:

0Y0000h	0Y2000h	0Y4000h	0Y6000h
0Y8000h	0YA000h	0YC000h	0YE000h

5.3 DIP Switch Settings

The AVT-1850-1 printed circuit board contains only one DIP switch. It does not have any other jumpers or switches. Refer to Section 2.1.1 for a detailed description on how to set the DIP switch for a selected segment address.

The first 7 switches on the DIP switch are used to set the board Segment address. The switches are connected to 10K ohm pull up resistors and are the input to a seven bit comparator. Switch #1 corresponds to the PC host address line A19; switch #2 - A18; ... switch #7 - A13. Therefore switch #1 is the MSB of the board segment address. When the host PC address lines match the settings of switches #1 through #7 of the DIP switch, the board is selected (on-board signal 'Board Select' goes active, low).

Switches #1 through #7 are factory defaulted to address B200,
OFF - ON - OFF - OFF - ON - ON - OFF (switches #1 through #7).

Switch #8 is connected to a pull up resistor and to the ALC line of the HIP7030A0 device. This switch should normally be set to ON (factory default). This keeps the ALC input of the HIP7030A0 device pulled low. ALC signal line should be low for 'normal' operation of the HIP7030A0 device. Consult the technical literature for the HIP7030A0 device for an explanation of the use of the ALC signal line.

Switch #9 is used to apply an approximate 10K ohm load to the external J1850 bus. SAE Specification J1850 states a J1850 bus node resistance to be a nominal 10.6K ohm. This switch is normally ON (factory default). It may be set to OFF to minimize bus loading during testing or for other reasons.

Switch #10 is not used and is not connected. It is factory defaulted to the OFF position.

5.4 HIP7030A0 Microcontroller

The Harris Semiconductor HIP7030A0 device is the heart of the AVT-1850-1. It consists of a 68HC05 microcontroller along with several highly integrated built in timer and filter functions. It is a dedicated and autonomous interface to the J1850 bus. In the network monitor role the HIP7030A0 serves to off-load from the host PC all of the tasks required to interface to the J1850 bus.

When the AVT-1850-1 is used as a development system for J1850 nodes, the HIP7030A0 microcontroller emulates the operation of the target device, the HIP7030A2 device.

The designer/developer should refer to Harris Semiconductor Technical literature for detailed information about the HIP7030A0, the HIP7030A2, and the related family of devices. Section 7.1 contains reference information.

5.4.1 Overview

As stated earlier the Harris HIP7030A0 device is functionally identical to the Harris HIP7030A2 device. The A2 contains mask ROM and would be used in a production application. The A0 is the emulator device for the A2 in that all of its ROM is mapped off-chip. It is for this reason that the A0 device is used in the AVT-1850-1 product. The ROM for the A0 device, in the AVT-1850-1, is simulated through the use of dual port RAM. For most of the memory space on the AVT-1850-1 only the PC has read/write access to the memory while the A0 device has

read only access to the that same space. Therefore, the developer can load code, test it, dump it, modify it, and reload it; all without having to handle any hardware (such as with EPROM's).

The HIP7030 devices contain a 68HC05 engine integrated with two bi-directional programmable ports, a programmable timer, a watchdog timer, on-chip RAM, a serial peripheral interface, and a J1850 VPW symbol encoder/decoder. Only the VPW symbol encoder/decoder is discussed here, all other functions are generic and the designer is free to define their use as the situation demands. The VPW symbol encoder/decoder unit is of concern since it is the mechanism through which communications with the J1850 bus are accomplished.

The VPW symbol encoder/decoder unit (known as the SENDEC unit) contains the necessary functionality to generate and decode J1850 VPW symbols. The SENDEC unit also contains a digital filter to minimize the effects of noise on the bus (noise pulses of less than 7 microseconds are filtered out). The SENDEC unit relieves the designer from the responsibilities of timing each of the symbols that must be generated when transmitting and likewise timing the received ones.

Use of the SENDEC unit is accomplished through a number of control registers that are accessed through registers (memory locations), all of which are mapped to page 0 (address: 0x00h through 0x1Fh). There are a number of subtleties in using both the SENDEC unit as well as the J1850 bus. Refer to the appropriate SAE and Harris HIP7030 documentation for detailed information.

5.4.2 Instruction Set

The instruction set for the HIP7030 is the same as for a standard 68HC05 microcontroller. Although the number of instructions in the set are relatively few, the flexibility of the instructions and the various addressing modes make them very powerful. This level of flexibility can also make it difficult for the uninitiated person to read someone else's code. Good commenting by the code author/developer cannot be emphasized enough.

There are several references that the user can consult for information about the instruction set. The Harris technical data on the HIP7030A2 device includes a brief section on the instruction set and the addressing modes. Someone who has programmed the 6805 (or similar devices) will find the data sheet useful as a reminder of the instructions, modes, and syntax.

However, others may find the book "Understanding Small Microcontrollers" to be much more useful (refer to section 7.1). Though the author uses another version of the 68HC05 in the examples, the explanations of the instruction set, the addressing modes, and the instruction set details in Appendix A make the book a valuable reference text to the programmer.

5.4.3 Programming

Developing code for the HIP7030 and testing and analyzing it while running on the AVT-1850-1 is simplified through the Enhanced On-line Software (EOS) development environment. When in EOS, the user can edit the source code, compile it, download it to the AVT-1850-1 board, control the execution of the code by the HIP7030A0 device, and monitor the contents of the on-board memory.

Several sample programs are included with the AVT-1850-1 Development System. They are located in the "\SAMPLES" sub-directory of the EOS installation directory. The source code files use the ".S" file spec. When the assembler is invoked, the resulting assembled file will have

a ".H" file spec. A source listing may be obtained by selecting the listing option when invoking the assembler, it will have a ".LST" file spec.

5.4.4 Operation

The HIP7030 device contains a number of specialized functions that are integrated into the microcontroller. Each of these functions is accessed and controlled, by the developer, through a set of command registers. All of the command registers are accessed via memory operations and are located on "Page Zero." Page 0 encompasses addresses from 0x00h through 0x1Fh. These registers reside on-chip (HIP7030) and are only accessible via the HIP7030 device. Each of the command registers may contain one or more functions and may be read only, write only, or some combination of both. The host PC cannot 'see' these command registers.

5.4.5 Overview of the HIP7030 Memory Organization

The following is an overview of memory organization of the HIP7030A0 and A2 devices. This information is presented primarily to highlight the differences between the two. The user of the AVT-1850-1 should carefully read the Harris literature for the A2 and A0 devices and note the differences presented here.

5.4.5.1 Addresses 0x00h through 0x1Fh

This memory consists of status, control, and data registers for the various integrated functions of the HIP7030 device. Consult the Harris literature for details on these registers.

There are no differences in this area between the A0 and A2 devices.

5.4.5.2 Addresses 0x20h through 0x4Fh

This is user ROM space in the A2 device. This space does not, essentially, exist on the A0 device and is not available to the developer using the AVT-1850-1.

5.4.5.3 Addresses 0x50h through 0xFFh

This is RAM space in both the A0 and A2 devices. This RAM space physically exists internal to the device. The top 64 bytes (starting at address 0xFFh and moving down) are available for use as the stack. The bottom 112 bytes are available to the user as general RAM. The user should use caution not to use RAM that is too high, otherwise a conflict of memory could occur between the user and the stack.

There are no differences in this area between the A0 and A2 devices.

5.4.5.4 Addresses 0x100h through 0x8FFh

This is user ROM space on the A2 device. For A0 device on the AVT-1850-1 this space exists as ROM and is referred to as RAM #3. (This space exists as RAM to the host PC which can load the memory as directed by the developer.)

There are, essentially, no differences in this area between the A0 and A2 devices.

5.4.5.5 Addresses 0x900h through 0x1BFFh

In the A2 device this space is marked as unused. For the A0 device on the AVT-1850-1 this space encompasses the remainder of RAM #3, which exists as ROM space to the A0 device. (This space exists as RAM to the host PC which can load the memory as directed by the developer.)

5.4.5.6 Addresses 0x1C00h through 0x1DFBh

In the A2 device this space is marked as unused. For the A0 device on the AVT-1850-1 this space is known as RAM #2 and is RAM space to the A0 device. (This space exists as ROM space to the host PC.)

5.4.5.7 Addresses 0x1DFCh through 0x1DFFh

In the A2 device this space is marked as unused. For the A0 device on the AVT-1850-1 this space contains the two board control registers and the two FIFO's. Refer to Section 5.2.6 and 5.2.7 for more information.

5.4.5.8 Addresses 0x1E00h through 0x1EFFh

In A2 device this space is marked as unused. For the A0 device on the AVT-1850-1 this space is known as RAM #1 and exists as ROM to the A0 device. (This space exists as RAM to the host PC which can load the memory as directed by the developer.)

5.4.5.9 Addresses 0x1F00h through 0x1FFFh

In the A2 device this is user ROM space and contains built in test code, built in test vectors, and interrupt vectors. For the A0 device on the AVT-1850-1 this space is known as RAM #1 and exists as ROM to the A0 device. (This space exists as RAM to the host PC which can load the memory as directed by the developer.)

There are, essentially, no differences in this area between the A0 and A2 devices.

5.5 Bus Transceiver

The AVT-1850-1 utilizes the Harris HIP7020 J1850 Bus Transceiver for the physical interface to the external J1850 bus. This device provides an isolation between the CMOS/TTL signals of the HIP7030A0 and the J1850 bus signals. During normal bus communications the HIP7020 transceiver provides controlled rise/fall time of the transmitted symbols (to minimize bus EMI) and has an internal receive symbol filter to minimize the affects of noise on the external bus.

Another feature of the HIP7020 device is the incorporation of an internal loopback feature. This function is accessible, via the PC, through Control Register #2, bit #5. The loopback signal is active low. When a logic '0' is written to bit #5, loopback on the HIP7020 is enabled. In this condition the HIP7030A0 device is effectively disconnected from the external J1850 bus. However, transmitted symbols are still routed through the waveshaper then through the bus receiver. The "received" symbol then shows up on the receive symbol pin and is routed to the HIP7030A0 for processing. In this mode testing can be conducted while still connected to the external J1850 bus, but without actually transmitting.

The HIP7020 also provides a number of protective features including: ground fault tolerance, ESD protection, prevention of latch-up, and reverse battery protection.

6. J1850

The following is a brief overview of the J1850 multiplex standard. This is not intended to be a tutorial nor is it a comprehensive treatment of the subject. The reader is encouraged to obtain and study, at a minimum, the J1850 specification as published by the SAE. Refer to Section 7.1 for information on how to obtain a copy of the specification.

6.1 Introduction

The Society of Automotive Engineers (SAE) has designed, published, and adopted what is known as Surface Vehicle Standard J1850 “Class B Data Communications Network Interface.” This standard defines the characteristics of the Physical layer and the Data Link layer of the ISO Open Systems Interconnect model. These characteristics include: voltages, timing, bit definitions, message construction, physical connectivity, capacitance, resistance, length, etc. The J1850 Specification permits two different implementations of a network.

One version of J1850 utilizes a single wire (plus ground), has a data rate of 10.4 k bits per second (kbps), and the symbols are defined by a combination of the voltage level and the pulse width. This version is known as Variable Pulse Width or VPW.

The second version of J1850 utilizes a two wire connection, has a data rate of 41.6 kbps, and the symbols are defined by a combination of timing rising edge to falling edge and rising edge to rising edge. This version is known as Pulse Width Modulation or PWM.

The AVT-1850-1 utilizes the HIP7030A0 device which implements the VPW version of J1850. Therefore the AVT-1850-1 is intended to be used with a J1850 VPW multiplex bus.

6.2 Operation

The design of the SENDEC unit of the HIP7030 devices relieves the developer of having to perform the actual timing of both transmitted and received symbols. The SENDEC also incorporates a receive symbol filter to filter out short duration noise pulses. The SENDEC has built in dedicated timer functions for symbol encoding and decoding as well as a dedicated interrupt with the 68HC05 microcontroller.

To transmit, the user writes the appropriate code to the SENDEC Data Register (SEDDR) and the SENDEC unit begins transmission of the specified symbol. All of the J1850 specified symbols are supported by the SENDEC unit. The SENDEC unit generates an interrupt when transmission of the latest symbol is about to be completed. Thus, transmission essentially only requires that the user wait until interrupted for the next symbol, write the desired symbol to the SEDDR, and then return from the interrupt until the SENDEC interrupts again for the next symbol.

Receive operation is likewise just as easy. When a symbol has been received, the SENDEC decodes the symbol, places the received symbol information in the SEDDR, and generates an interrupt. The user only has to answer the interrupt and read the SEDDR to obtain the latest received symbol. If noise is received, in place of a valid symbol, the SENDEC reports this as well, in the SENDEC status register, SEDSR.

It should be noted here that it may be possible to transmit or receive symbols using the HIP7030 device without using the SENDEC interrupt function but, rather, by using a polling method of looking at the various SENDEC control registers. However, this approach is not recommended due to timing constraints on symbol generation reception.

When transmission of a symbol has been commanded, the SENDEC unit generates a transition on the VPW OUT pin of HIP7030A0 device. This transition represents the start of a new symbol and it propagates through the HIP7020 bus transceiver. An echo of this transition is received back into the HIP7020, propagates through the filter and the bus receiver. The transition then appears on the RX pin of the HIP7020 and is routed to the !VPW IN pin of the HIP7030A0 microcontroller. This received transition is then processed by the SENDEC unit, is decoded and the results placed into the appropriate SENDEC control register(s) and an interrupt is generated (if enabled). By this time the user has 34 microseconds in which to evaluate the received symbol, decide what action to take, and take the necessary action. Possible actions may include generation of the next symbol, terminate transmission (in the event of loss of arbitration), or just receive and store the symbol.

The construction of a valid J1850 network message is, however, more complicated than just sending and receiving symbols. The specification spells out how a message shall be constructed, how transmitting nodes on a bus arbitrate for possession of the bus, use of a Cyclic Redundancy Check byte (CRC), and other important details. These requirements must be met by the node developer in the design of the embedded software and/or hardware. Neither the HIP7030 devices nor the AVT-1850-1 Development System automate these requirements.

How, or even if, the various requirements, as called out in J1850 (or other related specifications), are met is up to the designer/developer. Some insight may be gleaned from examining and/or experimenting with the sample code files included with the AVT-1850-1 Development System. The user is also referred to SAE publication SP-1070, “Automotive Multiplexing Technology” for information on implementing J1850 compliant code on the HIP7030A2 device. Refer to Section 7.1.

7. Technical

This section contains reference information on the J1850 specification, the various hardware used on the AVT-1850-1 board, information specific to the AVT-1850-1 Development System, and schematics of the AVT-1850-1 board.

7.1 References

The following materials are referenced in this manual. These documents should be consulted for additional or detailed information.

1. SAE Standard J1850 “Class B Data Communications Network Interface.” Available from the SAE, 400 Commonwealth Drive, Warrendale, PA 15096-0001; phone: 412-776-4970.
2. HIP7030A0 “J1850 8-Bit 68HC05 Microcontroller Emulator Version” data sheet. Available from Harris Semiconductor, PO Box 883, Melbourne, FL 32902-0883 phone: 800-442-7747.
3. HIP7030A2 “J1850 8-Bit 68HC05 Microcontroller” data sheet. Available from Harris Semiconductor, PO Box 883, Melbourne, FL 32902-0883 phone: 800-442-7747.

4. HIP7020 "J1850 Bus Transceiver I/O for Multiplex Wiring" data sheet.
   Available from Harris Semiconductor, PO Box 883, Melbourne, FL 32902-0883
   phone: 800-442-7747.
5. "Understanding Small Microcontrollers," by James M. Sibigtroth. Prentice Hall, Englewood Cliffs, NJ 07632. ISBN: 0-13-089129-0.
6. Automotive Multiplexing Technology. Papers from the 1995 SAE World Congress. SAE publication SP-1070.
   Available from the SAE, 400 Commonwealth Drive, Warrendale, PA 15096-0001;
   phone: 412-776-4970

7.2 Technical Support

The user may contact Advanced Vehicle Technologies, Inc. for assistance in any of the areas covered here. When calling please be prepared to identify yourself, tell us the serial number of your hardware, and the version number of EOS that you are using.

Advanced Vehicle Technologies, Inc. is located in Maryland and is open from 0800 hrs. to 1800 hrs. Eastern Time. If calling after hours, please leave a message and we will return your call as quickly as possible.

You may also fax your questions to us. We will either fax an answer back or call you, at your request. If faxing your question, please include as much relevant information about your question or problem as you can.

Our phone numbers are:

Voice: 410-798-4038
Fax: 410-798-4308
Beeper (with numeric, alpha, and voice mail): 800-770-7614

7.3 The AVT-1850-1 Board

The AVT-1850-1 board was designed to fit into and interface with a PC-AT or compatible computer. It utilizes an 8 bit ISA standard interface and is built using the full size AT board outline standard.

All of the board devices are CMOS or compatible. The board interfaces to the ISA bus using bus drivers and transceivers of the 74HCT family. The board requires +5 volts for all devices with the exception of the HIP7020 device which requires +12 volts. Both of these operating voltages are derived from the PC via the ISA bus interface. The use of all CMOS devices reduces to total board power dissipation to less than 700 milliwatts.

The designer/developer has access to all of the HIP7030 interfaces via a DB-25S connector mounted on the board edge and accessible when installed in the host PC. Refer to Section 7.4 for a list of pin definitions. Refer to HIP7030A2 technical literature for a detailed description of the signals and functions.

7.4 Connector Pinout

The following is a list of the pin functions on the board connector. The connector is a DB-25S and will mate with any industry standard DB-25P connector.

Pin Number	Name	Function / Description
1	GROUND	Connected to the board ground plane. Common for the J1850 Bus.
2	BUS	Connection to the external J1850 Bus.
3	!SS	Slave Select, input. Part of the Serial Peripheral Interface (SPI).
4	DS	Data Strobe, output. Only on the HIP7030A0 device.
5	MISO	Master In, Slave Out, input/output. Part of the Serial Peripheral Interface (SPI).
6	SCK	Serial Clock, input/output. Part of the Serial Peripheral Interface (SPI).
7	PD4	Port D bit 4, input/output. Programmable.
8	PD2	Port D bit 2, input/output. Programmable.
9	PD0	Port D bit 0, input/output. Programmable.
10	PA6	Port A bit 6, input/output. Programmable.
11	PA4	Port A bit 4, input/output. Programmable.
12	PA2	Port A bit 5, input/output. Programmable.
13	PA0	Port A bit 0, input/output. Programmable.
14	!IRQ	Maskable Interrupt Request, input.
15	TCAP	Timer Capture, input. Used with the programmable timer feature or with Port D.
16	OSCB	Oscillator Buffered Output. Buffered output of the version of the OSCIN signal. On the AVT-1850-1 it is a 10 MHz clock signal.
17	FS	Fetch Status, output. Only on the HIP7030A0 device.
18	MOSI	Master Out, Slave In, input/output. Part of the Serial Peripheral Interface (SPI).
19	TCMP	Timer Compare, output. Used with the programmable timer feature.
20	PD3	Port D bit 3, input/output. Programmable.
21	PD1	Port D bit 1, input/output. Programmable.
22	PA7	Port A bit 7, input/output. Programmable.
23	PA5	Port A bit 5, input/output. Programmable.
24	PA3	Port A bit 3, input/output. Programmable.
25	PA1	Port A bit 1, input/output. Programmable.

8. Appendix A - Contents of the Distribution Disk

The installation disk for EOS contains the following files. These files are copied, during installation, to the hard disk drive of the host PC during installation.

Contents of \EOS directory:

INSTALL.EXE - The executable that creates the destination directory for the EOS program and files, unzips the EOS program and support files, and updates the AUTOEXEC.BAT file.

EOS.EXE - The EOS executable that provides the integrated environment for using the AVT-1850-1.

FIFO.H - The AVT-1850-1 program used by EOS.EXE to run FIFO diagnostics. This file must stay in the destination directory in order to run diagnostics.

HASM.DOC - User's documentation for the Harris 6805 assembler.

HASM.EXE - Harris 6805 assembler supplied for the EOS program. This file must stay in the destination directory in order for EOS to have the capability of assembling the users programs.

INIT.H - Bootstrap program run by EOS.EXE on the program's startup. This file must be in the destination directory in order to automatically bring the AVT-1850-1 up into a "good" state.

RAM_1.H - The AVT-1850-1 program used by EOS.EXE to run RAM1 diagnostics. This file must stay in the destination directory in order to run diagnostics.

RAM_2.H - The AVT-1850-1 program used by EOS.EXE to run RAM2 diagnostics. This file must stay in the destination directory in order to run diagnostics.

RAM_3.H - The AVT-1850-1 program used by EOS.EXE to run RAM3 diagnostics. This file must stay in the destination directory in order to run diagnostics.

RAM_4.H - The AVT-1850-1 program used by EOS.EXE to run RAM4 diagnostics. This file must stay in the destination directory in order to run diagnostics.

MONITOR.H - The AVT-1850-1 program used while in the Network Monitor mode. This program is automatically downloaded onto the AVT-1850-1 board and execution begun when the START/UPDATE button on the Network Monitor window is pressed.

MONITOR.S - The assembly language source code for the MONITOR.H program.

AUTOEXEC.EOS - This file is created by the installation program and contains the environment variables and the path declarations needed in order to run the EOS program. The installation program will update the AUTOEXEC.BAT file automatically with these values; however, if the user had requested that the program not do the automatic update, the lines in this file may be inserted manually.

In addition to the above files, the destination directory has two sub-directories: DIAG and SAMPLES. The DIAG sub-directory contains the source code for the diagnostic programs as well as the bootstrap program. These are here for educational reasons and should not be modified by the user.

Contents of the \EOS\DIAG directory:

FIFO.S - Source for the FIFO diagnostic program.

INIT.S - Source for the EOS bootstrap program.

RAM_1.S - Source for the RAM1 diagnostic program.

RAM_2.S - Source for the RAM2 diagnostic program.

RAM_3.S - Source for the RAM3 diagnostic program.

RAM_4.S - Source for the RAM4 diagnostic program.

TEST.TXT - Notes for the test programs.

The second sub-directory, SAMPLES, contains sample programs to assist the user in developing AVT-1850-1 software. These programs are offered without guarantees and should be viewed for educational purposes only.

Contents of the \EOS\SAMPLE directory:

CRC.S - Routine to calculate the CRC byte.

MEM_MAP.S - Contains the memory map of the AVT-1850-1 memory functions and locations.

NOTES1.TXT - Text file containing useful notes for the RCV_1, RCV_2, RCV_3, X&R_1, X&R_2, XMIT_1, XMIT_2, and XMIT_3 programs.

RCV_1.S - A program to only receive information from the J1850 bus.

RCV_2.S - A second program to only receive information from the J1850 bus.

RCV_3.S - A third program to only receive information from the J1850 bus.

X&R_1.S - A program that will transmit and receive a message on the J1850 bus.

X&R_2.S - A second program to transmit and receive messages on the J1850 bus.

XMIT_1.S - A program to only transmit a message.

XMIT_2.S - A second program to only transmit a message.

XMIT_3.S - A third program to only transmit a message.

X_AUTO.S - Source code for an automatic transmit program.

NOTES2.TXT - Notes on the use of the X_AUTO and MONITOR programs.

9. Appendix B - Description of the EOS Installation Program

The installation executable “INSTALL.EXE” is a menu driven program that creates the destination directory, copies the EOS program and support files, and updates the AUTOEXEC.BAT file with EOS specific information. In order to run this program the user must insert the installation disk in either drive “A” or drive “B” and type “A:\INSTALL.EXE” or “B:\INSTALL.EXE”, respectively.

Once the installation program has started, the user is given a menu bar with two options: File and Help. Both of these menu options have menu items that can be selected. These menu items are described in the following paragraphs.

The first File menu item, START INSTALLATION, brings up the EOS Installation dialog box. This dialog box is the driver for the installation program. From it the user selects the destination (drive and directory where the EOS software is to be installed), the source (the drive containing the installation disk), the Segment Specification, the number of AVT-1850-1 boards being installed in the PC chassis, and the automatic update of the AUTOEXEC.BAT file. The segment selection option, which is described in section 2.1.1, gives the user the option of having the software search for a free segments in PC-DOS memory or to let the user specify a known good segment. The update of the AUTOEXEC.BAT file option permits the installation program to automatically update the AUTOEXEC.BAT file with the proper environment and path variables. (Note the installation program assumes that the AUTOEXEC.BAT file is located on the "C:" drive.) Once the proper options have been entered, the user presses the "OK" to start the installation.

The second File menu item, EXIT, terminates the installation program. This is the only method for exiting the installation program.

The Help menu option only has one menu item: ABOUT. This option brings up the About EOS Installation dialog box which contains information about the installation program.

Once the installation program has completed, the user should reboot the PC in order to permit any environment variable updates to take effect.

10. Company Overview

Advanced Vehicle Technologies, Inc. is dedicated to providing affordable hardware, software, and technical support to the developers and users of vehicle based multiplex networks. The AVT-1850-1 represents our implementation of the VPW version of the SAE J1850 multiplex bus standard.

AVT also offers other vehicle multiplex bus products including an interface to the J1850 PWM multiplex bus standard for a host PC or laptop machine.

The engineering staff at AVT has combined experience on a number of multiplex bus standards including: J1850 VPW and PWM and ISO-9141 and 9141-2. Members of the staff are available to provide assistance on the use of any of AVT's products.

AVT engineering staff members are available to provide dedicated engineering support for a customer project. Through a simple contractual arrangement, a customer is able to 'tap' into AVT's knowledge and experience base.

Information on any of the products or engineering support that Advanced Vehicle Technologies can provide is available by calling, faxing, or writing.

Advanced Vehicle Technologies, Inc.
1509 Manor View Road
Davidsonville, MD 21035
410-798-4038 (voice) 410-798-4308 (fax)