ADVANCED VEHICLE TECHNOLOGIES, Inc. AVT inc.

AVT-717

UBP / CAN Interface

RS-232/422 Unit

User's Manual

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

Revision 'A'
February 2007

TABLE OF CONTENTS

1. INTRODUCTION

This manual covers the AVT-717 UBP / CAN Interface unit. It provides technical information on using the interface, connections to it, communications with it, and other related information.

The AVT-717 UBP / CAN Interface supports the following modes of operation.

- Ford UBP (UART Based Protocol).
- CAN (Controller Area Network).

The **AVT-717 Hardware Only** package (order number: 717-002) consists of the AVT-717 UBP / CAN Interface board housed in rugged polycarbonate enclosure; a serial cable; and this User's Manual. All connections to the enclosure are identified in Section 6.4.

The **AVT-717 OEM** module (order number: 717-003) consists of the AVT-717 UBP / CAN Interface board only. All connections to the interface board are identified in Section 6.3.

Operations in UBP mode conforms to the Ford Motor Company Standard UBP standard.

Operations in CAN mode are compatible with ISO-11898, J1939, and other specifications.

The AVT-717 UBP / CAN Interface provides the following functions:

- Isolated electrical interface between the network and the control computer.
- Protocol/data conversion between a vehicle based network and the host computer (via a serial communications link).
- Passive network traffic monitor.

1.1 Specifications and Requirements

<u>AVT-717 UBP / CAN Interface (in enclosure)</u>

- Overall size (inches): 6.7 wide x 2.2 high
  x 4.75 deep (5.4 including switch and connectors).
- Weight: 15 oz.
- +12 volts (nominal) from subject vehicle.
- Input voltage range: +8.5 VDC to +24 VDC.
- Power dissipation: 2.5 watts (nominal).

<u>AVT-717 UBP / CAN Interface (OEM board)</u>

- Overall size: 4.0 wide x 5.6 deep x 0.8 high (inches).
- Weight: 4 oz.
- +12 volts (nominal) from subject vehicle.
- Input voltage range: +8.5 VDC to +24 VDC.
- Power dissipation: 2.5 watts (nominal).

1.2 Definitions

The following terms are used in this manual.

- UBP: UART Based Protocol (Ford Motor Company proprietary).
- CAN: Controller Area Network.
- SWC: Single Wire CAN (General Motors unique).
- All numbers used in this manual are hexadecimal digits (0 .. 9 and A .. F) and are usually preceded with a dollar sign ($) for clarity.

2. Installation

Prior to using the AVT-717 the type and baud rate of the serial link between it and the host computer must be properly set. The interface must then be connected to both the subject vehicle and the control computer.

2.1 Hardware Configuration

The AVT-717 serial interface can be configured for either RS-232 or RS-422 operation . Additionally, the data rate can be configured for 9.6k, 19.2k, 38.4k, or 57.6k baud rate. These configuration selections are set by jumpers and connectors on the AVT-717 UBP / CAN Interface board. (Note: documentation by other manufacturers may make reference to 56k baud. This is usually the same as what we refer to as 57.6k baud.)

The factory default settings for the AVT-717 are for RS-232 operation at 9.6k baud rate.

Refer to the following instructions and Figures 1 and 2 for details on configuring the AVT-717 serial communications to the host computer.

<u>To select RS-232 operation:</u>

- Locate jumper JP1 on AVT-717 board.
- Place the jumper across pins 3 and 5 on JP1.
- Place the jumper across pins 4 and 6 on JP1.
- Connect the internal serial cable to P1 on AVT-717 board.
  (The internal serial cable is a ribbon cable connected to the enclosure mounted connector P2, it’s the narrower of the two cables.)

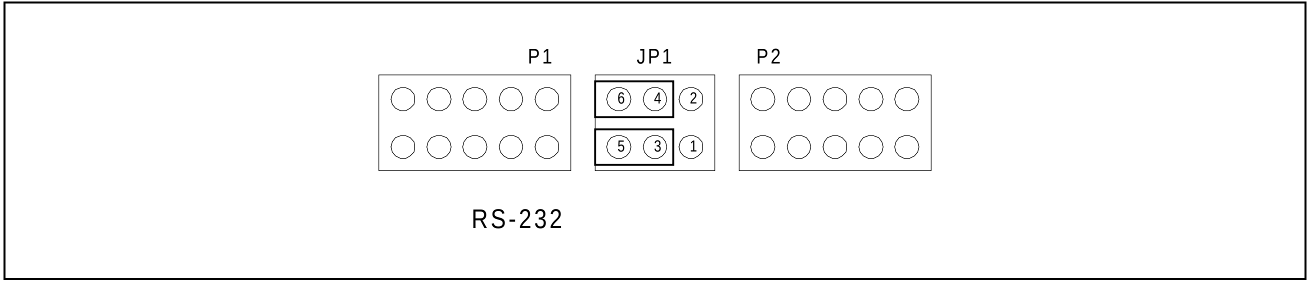

Figure 1 RS-232 Connector and Jumper Configuration

To select RS-422 operation:

- Locate jumper JP1 on AVT-717 board.
- Place the jumper across pins 1 and 3 on JP1.
- Place the jumper across pins 2 and 4 on JP1.
- Connect the internal serial cable to P2 on AVT-717 board.
  (The internal serial cable is a ribbon cable connected to the enclosure mounted connector P2, it's the narrower of the two cables.)

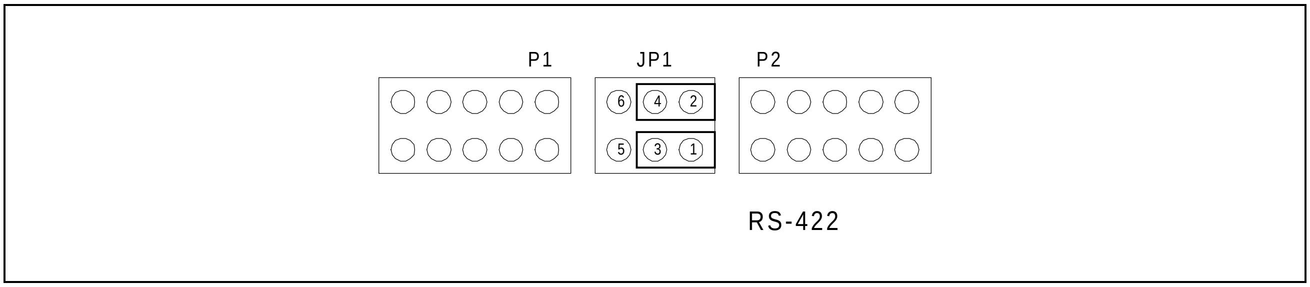

Figure 2 RS-422 Connector and Jumper Configuration

To set the baud rate:

The baud rate is determined by installing or removing the jumpers on JP3 and JP4 on the AVT-717 board. Refer to Tables 1 and 2 for JP3 and JP4 settings and corresponding baud rates for the different firmware versions.

Baud Rate	JP3	JP4
9.6k	In	In
19.2k	Out	In
38.4k	In	Out
57.6k	Out	Out

Table 1 Baud Rate Selection (firmware v2.0 and below).

Baud Rate	JP3	JP4
19.2k	In	In
38.4k	Out	In
57.6k	In	Out
115.2k	Out	Out

Table 2 Baud Rate Selection (firmware v2.2 and above).

2.2 Hardware Connections

Initial use and testing require that the AVT-717 be connected to a power supply and the host computer serial port. Consult Section 6.3 or 6.4 for the pin assignments of the connectors on the enclosure and the pin assignments of the connectors on the OEM board.

The external power source should be a nominal +12v (minimum of +8.5v and maximum of +24v).

It is recommended that the serial cable (provided with the hardware only package) be used during initial testing. Serial communications between the AVT-717 and the host computer utilize RTS/CTS hardware handshaking so all relevant pins must be connected.

No other equipment or connections are required to get the unit up and running and establish communications between the AVT-717 and the host computer.

2.3 Testing the AVT-717

Once the hardware connections have been made, proper operation of the hardware can be observed.

- Use the AVT Controller software or a similar communications program.
  Note that the AVT-717 communicates with the host computer using binary bytes (not ASCII digits).
- Set the parameters of the communications program appropriately (as determined by the configuration of the AVT-717 interface).
- Turn on the AVT-717 interface unit.
- Observe on the AVT-717 board (top) that the green (power) LED is lit.
- Observe that the red (operations) LED is blinking fast.
- Observe on the computer that the following is received from the interface:
  91 12 and 92 04 xx.
  This is the 'mode of operation' report. $91 indicates that the message is a 'board status report' with one byte to follow. The $12 indicates the interface is up and running at idle awaiting a mode switch command. The $92 $04 $xx is the firmware version report, with the firmware version being 'xx.' Refer to the Master Commands and Responses document for complete and up-to-date information.
  Note: The Controller software only displays and accepts hex digits, no dollar signs.
- At this point the AVT-717 hardware is operational.

3. Interface Introduction

The AVT-717 UBP / CAN Interface is available housed in a rugged polycarbonate enclosure with internal cable assemblies and power switch. The two connectors on the enclosure are for connections to the host computer, power, and the vehicle network.

The AVT-717 provides an isolated electrical interface between the control computer and the network under test. The AVT-717 Interface can be connected directly to a vehicle and derives its operating power from the vehicle. The serial interface to the control computer is electrically isolated from the vehicle electrical and electronic systems. This is done to prevent damage to the control computer due to spikes or surges from the vehicle electrical system. It is not recommended that the control computer be connected directly to the vehicle electrical system.

The AVT-717 UBP / CAN Interface performs the necessary data and protocol conversion functions.

When in UBP mode of operation a UPL (UART Physical Layer device) is utilized.

CAN operations are supported by the Intel 82527 CAN Controller device.

When in CAN mode there are two physical layer implementations available: GM's SWC (Single Wire CAN) and the more typical 2-wire CAN. In SWC mode the Infineon transceiver device is used. In 2-wire mode the Philips 82C250 transceiver is used.

All firmware on the AVT-717 UBP / CAN Interface was developed by and is supported by Advanced Vehicle Technologies, Inc.

The serial communications function is provided by a Philips Serial Communications Controller (SCC) device. The RS-232 and RS-422 line drivers and receivers are contained in an electrically isolated section of the AVT-717 board. Signals are coupled through optical isolators and power is supplied by an isolated DC-DC converter.

The AVT-717 UBP / CAN Interface utilizes a Motorola CPU32 core microcontroller. All firmware is contained in an EPROM device socket mounted on AVT-717 board. The board was designed to support the three modes of operation: UBP, CAN - SWC, and CAN 2-wire. The selection of operational mode is set by a software command from the control computer to the interface unit.

Refer to Sections 5.4 and 5.5 for detailed information on operations in each of the supported modes.

4. Interface Description

A block diagram of the AVT-717 UBP / CAN Interface is shown in Figure 3. The heart of the unit is the Motorola 68332 microcontroller. This device utilizes a 68000 core with a bus speed of 16.777 MHz. Operation firmware is contained in an EPROM. Software commands from the control computer control the selection of the operational mode of the interface.

The microcontroller utilizes several peripheral devices: two FIFO's each 2 KBytes deep, the CAN controller, the serial communications controller, and a free running 23-bit microsecond clock.

The two FIFO's are used by the microcontroller as buffers for the transmit and receive operations between itself and the control computer. A serial communications controller is used to implement the standard serial data link, including RTS/CTS hardware handshaking. The serial communications controller is interfaced to the control computer through a signal isolation block and either an RS-232 or an RS-422 transceiver block.

The control computer serial data link is electrically isolated from the rest of the Interface unit and, hence, the vehicle through an isolated DC-to-DC power converter and a signal isolation block.

The CAN controller provides all the CAN functions. It is connected to the two external CAN buses by two transceivers; selected by software. One CAN transceiver is for the 2-wire implementation and the other transceiver is for the Single Wire CAN (SWC) implementation.

One half of the serial communications controller, the UPL device, and custom firmware provide the UBP interface function.

5. AVT-717 Operation

The following describes the use of the AVT-717 UBP / CAN Interface. It is assumed that the interface is properly connected to an external vehicle network and to the control computer. Furthermore, it is assumed that the AVT Controller software (or similar communications software) has been installed on the control computer, that it is running, and that the test described in Section 2.3 has been completed successfully.

5.1 Indicators

The AVT-717 board has two indicator LED's: one green and one red.

The green LED is connected to the +5 VDC supply for the board and provides a quick indication that power is available for normal operation. If the green LED should fail to light, check the power source from the subject vehicle, and check fuse F1 on the AVT-717 board.

The red LED is a heartbeat indicator. The microcontroller toggles the state of the red LED every 62.5 milliseconds. As a result the red LED flashes noticeably during normal operations. If a problem with the microcontroller should occur the LED will either go to a full ON or full OFF state. This should be readily apparent and be indicative of an abnormal condition.

5.2 Communications, Electrical

All communications between the AVT-717 Interface and the external network were designed to be in conformance with all relevant standards and specifications.

All communications between the AVT-717 UBP / CAN Interface and the control computer conform, at the physical interface, to either EIA-RS-232 or RS-422 standards (as selected by the user). Communications between the interface unit and the control computer follow industry standard serial communications protocol. There is one start bit, eight data bits (least significant bit first), no parity, and one stop bit. Additionally, hardware handshaking using the RTS/CTS signal lines is used.

When in RS-232 mode the output data line, TXD, idles low (-8 volts) and the RTS signal line idles high (+8 volts). Likewise, the AVT-717 expects RXD to idle low and CTS to idle high. (All voltages measured with respect to the communications isolated ground.) The user should be aware of potential communications errors that may occur when using RS-232 at the higher data rates or in an electrically noisy environment.

When in RS-422 mode the output data lines idle such that TXD+ is high (+4 volts) and TXD- is low (0 volts). The RTS+ signal idles low (0 volts) and RTS- is high (+4 volts). Likewise the AVT-717 expects RXD+ to idle high (RXD- is low) and CTS+ is low (CTS- is high). (All voltages measured with respect to the communications isolated ground.)

5.3 Communications, Messages

The structure and protocol of communications between the control computer and the interface are stated in the following sections.

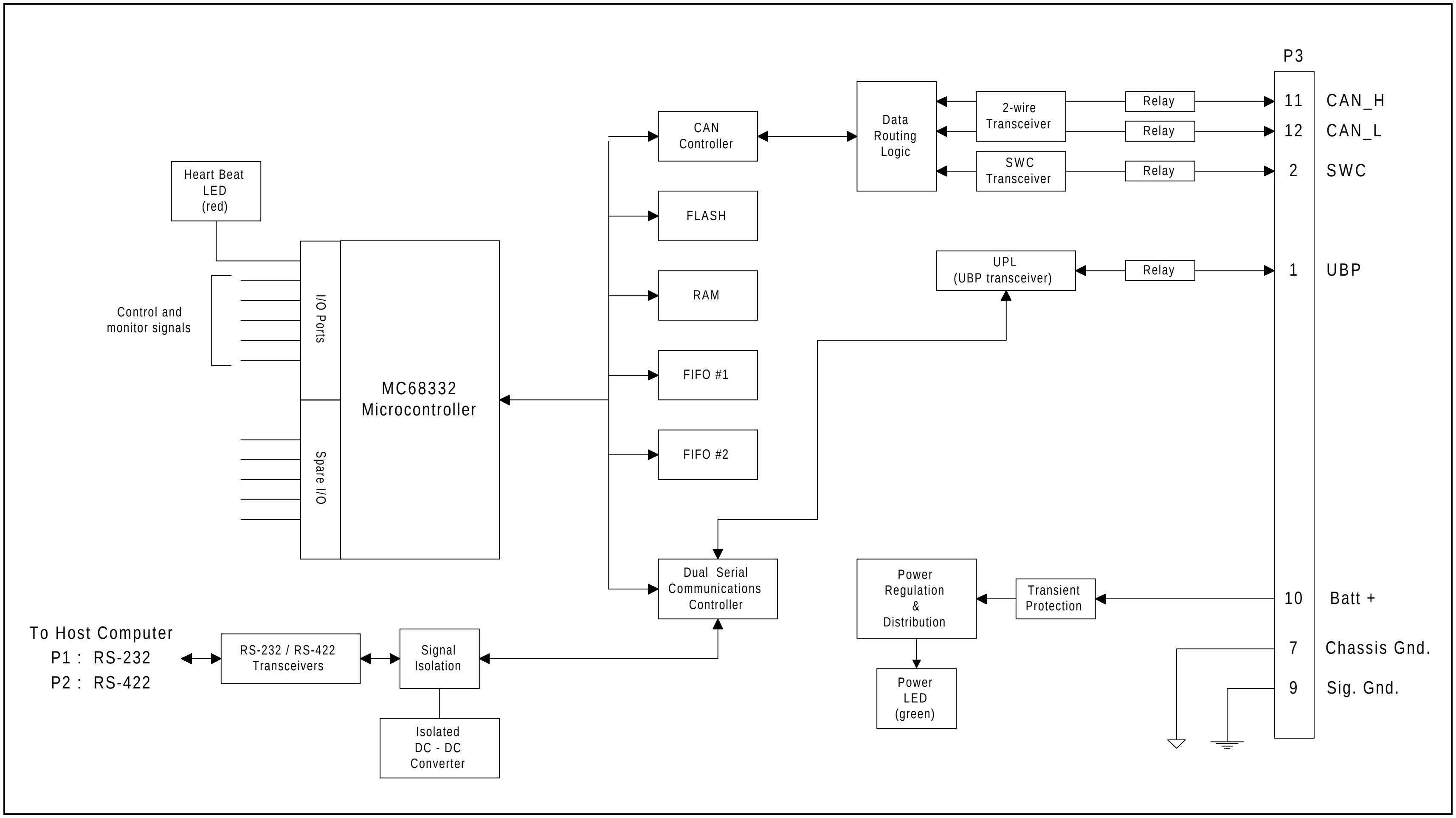

Figure 3 AVT-717 Unit Block Diagram

5.3.1 Packet Construction

Messages or packets from the host computer to the AVT-717 Interface are known as Commands.

Messages or packets from the AVT-717 Interface to the host computer are known as Responses.

All data is transferred in packets. The size of each data packet varies depending on the mode of operation. For most modes the packet length is from 1 byte to 16 bytes (inclusive).

The first byte in each data packet is the header byte and is used to convey information only between the control computer and the microcontroller in the interface.

The header byte is divided into the upper nibble and lower nibble. The upper nibble indicates what information the data packet is conveying. The lower nibble is the count of the number of bytes that follow the header byte. The meaning of the upper nibble of the header byte depends on which direction the data packet is moving; whether to or from the control computer.

There are occasions where packets between the AVT-717 and the host computer are more than 15 bytes in length. To support these longer packets, two alternate forms have been developed. Please consult the document "Long Messages - Alternate Header Formats."

5.3.2 Alternate Header Format

To accommodate long messages, the format for communications between the AVT-717 unit and the host computer has been modified. To send a message onto the bus the host has a choice of how to send the message to the 716 unit. These are described here, along with examples.

If the message is 15 bytes or less in length (total) then the 'normal' format may be used.

Normal Format: 0x aa bb cc ...
x - count of bytes to follow
aa bb cc ... message bytes.

Example: 05 81 F1 C1 48 9B

If the message is more than 15 bytes but less than 255 bytes in length, alternate format #1 is available using a header byte of $11.

Alternate format #1: 11 xx aa bb cc ...
11 - header byte
xx - count of bytes to follow
aa bb cc ... message bytes.

If the message is more than 255 bytes in length, alternate format #2 is available using a header byte of $12.

Alternate format #2: 12 xx yy aa bb cc ...
12 - header byte
xx - count of bytes to follow, most significant byte
yy - count of bytes to follow, least significant byte
aa bb cc ... message bytes.

These formats are backward compatible and may be used as desired.

For example, the host wants to transmit the following message onto the bus:
A1 B2 C3 D4 E5 F6 A7 B8 C9 DA EB FC AD BE

The following messages from the host to the AVT-717 are all equivalent. The header byte(s) have been bolded and underlined for clarity.

<u>0E</u> A1 B2 C3 D4 E5 F6 A7 B8 C9 DA EB FC AD BE

<u>11 0E</u> A1 B2 C3 D4 E5 F6 A7 B8 C9 DA EB FC AD BE

<u>12 00 0E</u> A1 B2 C3 D4 E5 F6 A7 B8 C9 DA EB FC AD BE

5.4 Firmware Notes

Firmware version 2.0 and below offer CAN and UBP modes only.

Firmware versions 2.2 and above offer CAN, UBP, LIN, and ALDL modes.

5.5 CAN Mode

Consult the latest version of the “Commands and Responses” list for detailed information on the commands supported by the AVT-717 while in CAN mode of operation.

The latest version of the “Commands and Responses” document can be obtained from our web site at: www.AVT-HQ.com

Please refer to the document: “A Discussion about CAN” for even more information on how to use the AVT-717 in CAN mode.

5.5.1 ISO 15765 Support

This may also be known as Segmented or Multi-Frame messaging.

Firmware version 2.2 and above provide full ISO 15765 support. Refer to the “Master Commands and Responses” document for detailed information of this capability.

5.6 UBP Mode

Consult the latest version of the “Commands and Responses” list for detailed information on the commands supported by the AVT-717 while in UBP mode of operation.

The latest version of the “Commands and Responses” document can be obtained from our web site at: www.AVT-HQ.com

When in UBP mode of operation the interface is always listening to, or monitoring, the external network. All bus traffic is reported to the control computer by the interface. Transmit operations occur only when initiated by the control computer.

5.6.1 Transmit message format

To transmit a message onto the network the message must be built by the operator and then sent to the AVT-717. The checksum byte will automatically be calculated and appended onto the message when transmitted. This can be enabled or disabled by software command.

It is up to the user to determine and know the proper messaging strategy that is used on the external network to which the AVT-717 is attached.

Any message destined for transmission must be preceded by a byte whose upper nibble is '0' (zero) and lower nibble is the byte count of the message (unless one of the alternate header forms is used). The message bytes then follow immediately.

Each time a message transmission is attempted the interface will issue a $A1 $xx status report indicating what happened. Refer to the Master Commands and Responses document for a description of the bit map for the $xx status byte.

5.6.2 Transmit Message Status Byte Definitions

The received message status byte always follows the header byte, even if the status byte is the result of transmitting a message.

Bit	Definition
0:	
1:	Framing error.
2:	Break received.
3:	
4:	Lost arbitration.
5:	Transmission successful.
6:	
7:	Arbitration limit exceeded (5 attempts).

5.6.3 Receive message format

Messages received from the network are assembled into the original byte sequence. The received checksum is calculated and checked to be equal to the checksum at the end of the message. The received checksum byte is discarded or preserved according to command $52 $01 $xx.

The Format byte (first byte) is also checked to match the actual message, specifically the length of the data field.

Status of these two tests are indicated in the received status byte.

The received message is then forwarded to the host in the following format:
Header byte - indicating message from the network and number of bytes to follow.
Time stamp - 4 bytes, microsecond resolution, if function is enabled.
Status byte - bit map of message status.
Message bytes - follow immediately.
Checksum byte - if function is enabled.

As an example the byte sequence $A7 $B6 $C5 plus checksum byte is transmitted by a node. The transmitted message is received by the interface and the following byte sequence is passed to the control computer: $04 $00 $A7 $B6 $C5 (time stamping is disabled).

The byte $04 indicates that it is a received message and that four bytes follow.
The byte $00 is the received message status byte and indicates that no errors were found.
(Received message status byte, bit definitions are listed in the next section.)
The message bytes then follow. Note that the CRC bytes are stripped off.

5.6.4 Received Message Status Byte Definitions

The received message status byte follows the header byte or the time stamp.

Bit	Definition
0:	Checksum error.
1:	Framing error.
2:	Break received.
3:	
4:	
5:	
6:	From this device.
7:	Message length does not equal Format byte.

5.6.5 Examples

To illustrate the construction and decoding of messages between the control computer and the AVT-717 interface several examples are provided.

Example #1: Want to request the current operational mode.
Command string: D0.
The interface responds with: 91 14. The ‘9’ indicates a board response, the ‘1’ indicates one byte follows, and the 14 indicates UBP mode.

Example #2: want to send a message out on the bus.
Command string: 04 32 89 AC 5F.
The interface responds with: 01 40. The ‘0’ indicates a received message and the ‘1’ indicates only one byte, which is the received message status byte. The ‘40’ indicates that bit 6 is set which means the received message was from this device. (Messages transmitted by the interface are received by the interface, are checked for errors, but are only the status byte is sent to the host, unless otherwise commanded by the $52 $04 $xx command.) The AVT-717 will also send the transmission acknowledgment A1 60 which indicates that a message was successfully transmitted.

5.7 LIN Mode

Consult the latest version of the “Master Commands and Responses” document for detailed information on the commands supported by the AVT-717 while in LIN mode of operation. Also included in that document are transmit and receive message formats as well as receive status byte definitions.

The latest version of the "Commands and Responses" document can be obtained from our web site at: www.AVT-HQ.com

5.8 GM +12v ALDL or 8192 UART Mode

The AVT-717 firmware version 2.2 and above supports the GM +12v version of ALDL or 8192 UART mode.

Consult the latest version of the "Master Commands and Responses" document for detailed information on the commands supported by the AVT-717 while in LIN mode of operation. Also included in that document are transmit and receive message formats as well as receive status byte definitions.

The latest version of the "Commands and Responses" document can be obtained from our web site at: www.AVT-HQ.com

5.9 Match Function

A coarse filtering mechanism for messages received from the bus is provided by the AVT-717 Interface unit firmware. *The match function is NOT applicable to CAN mode of operation.* If the match table is cleared (on power-up, reset, or $31 $7B command) all messages received from the network are passed to the host.

When at least one entry is made to the match table, all messages received from the network are checked against the match table. If a match is found the message is passed to the host. If no match is found, the message is discarded, and the host is not notified.

A match table entry is made using the $32 $xx $yy command. The $xx value is the byte position and the $yy value is the byte value. This filtering mechanism is more easily explained by example.

> It is desired to receive all messages (at the host) where the third byte of the message is equal to $F1. Send the command $32 $03 $F1 to the AVT-717 interface. To verify the table entry send the command $30. The response will be $42 $03 $F1. The only network messages passed to the host will now be of the form: $zz $xx $F1 $... Note that at the host the message will be $rr $ss $zz $xx $F1 $.. where $rr is the header byte, $ss is the received message status byte, and the message follows.

The match table can hold ten entries where an entry consists of a byte position and a byte value. The byte position refers to where in the network message the match byte is to be compared. The first byte of the message has a byte position value of one.

Ordering of the match table is not important. All table entries are checked until a match is found or the end of the table is encountered. If a match table entry specifies a byte position that doesn't exist for the message being checked (the message is shorter than the table entry), that table entry is not checked.

Note that the header byte and the received message status byte are not included in the match function nor are these two bytes considered part of the message. When in UBP and ISO modes the byte following the header byte is the received message status byte. When in PWM mode the byte following the header byte is the message number. These bytes are ignored by the match function and are not counted. The very next byte is the first byte of the message and has is byte number one of the message.

5.10 Status and Error Codes

Consult the latest version of the "Commands and Responses" list for a complete list of Status and Error codes supported by the AVT-717.

The latest version of the "Commands and Responses" document can be obtained from our web site at: www.AVT-HQ.com

6. Technical Information

6.1 Reference Documentation

This section contains reference information.

1.
2.

6.2 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 and tell us the serial number of your hardware.

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.

We can be contacted:

Voice:	410-798-4038
Fax:	410-798-4308
or by E-mail at:	support@AVT-HQ.com

6.3 AVT-717 UBP / CAN Interface Information

Information about the enclosure is contained in Section 6.4.

The following sections contain information about the AVT-717 UBP / CAN Interface board.

6.3.1 Fuse F1

Fuse F1 on the AVT-717 board is a 500 milliamp fast blow fuse designed to protect the AVT-717 in the event of reverse voltage application or a voltage surge significant enough to trip the input transient voltage suppressor.

The fuse is a Schurter # MSF 125 034.4216.

6.3.2 Jumper JP1

JP1 on the AVT-717 board is used to select which serial communications port is active, RS-232 or RS-422. The jumpers on JP1 should be set as noted in Section 2.1.

6.3.3 Jumper JP2

JP2 is a three position jumper used to select the UBP bus pull-up resistor. Only one jumper should be installed at a time. The following table shows the available pull-up values and jumper position.

Jumper Position	Pull-up resistor value
1 - 2	75 K ohm
3 - 4	18.2 K ohm
5 - 6	2.74 K ohm

6.3.4 Jumpers JP3 and JP4

JP3 and JP4 on the AVT-717 board are used to select the serial communications port baud rate. Set the jumpers for the desired baud rate in accordance with the instructions in Section 2.1.

6.3.5 Connector P1

P1 on the AVT-717 board is the RS-232 serial port connection. The connector is a 10 position header with pins on 0.100 inch centers. The header is compatible with an AMP #111917-1 a 10 position IDC ribbon cable connector.

Only the indicated pins are connected, all others are ‘not connected.’

Pin Number	Name		
1	CD	+5v (via a 10K ohm resistor)	output
2		connected to pin #7 (for DTR/DSR loop)	
3	TXD	Transmit Data	output
4	CTS	Clear To Send	input
5	RXD	Receive Data	input
6	RTS	Ready To Send	output
7		connected to pin #2 (for DTR/DSR loop)	
9	Sig. Gnd.	Isolated from vehicle ground	

6.3.6 Connector P2

P2 on the AVT-717 board is the RS-422 serial port connection. The connector is a 10 position header with pins on 0.100 inch centers. The header is compatible with an AMP #111917-1 a 10 position IDC connector.

Only the indicated pins are connected, all others are 'not connected.'

Pin Number	Name	Function / Description	
1	Sig. Gnd.	Isolated from vehicle ground	
2	RTS -	Ready To Send (inverted)	output
3	CTS +	Clear To Send	input
4	RTS +	Ready To Send	output
5	CTS -	Clear To Send (inverted)	input
6	TXD +	Transmit data	output
7	RXD +	Receive data	input
8	TXD -	Transmit data (inverted)	output
9	RXD -	Receive data (inverted)	input

6.3.7 Connector P3

P3 on the AVT-717 board is the connection to external power and external networks. The connector is a 16 position header with pins on 0.100 inch centers. The header is compatible with an AMP #111918-3 a 16 position IDC connector.

Only the indicated pins are connected, all others are 'not connected.'

Pin #	Signal
1	UBP bus.
2	SWC CAN bus.
7	Ground (chassis).
9	Ground (signal).
10	External power (+12v nominal).
11	CAN - H bus.
12	CAN - L bus.

6.3.8 Connector P4

P4 on the AVT-717#2 board enables or disables the high side supply voltage to the entire AVT-717 UBP / CAN Interface board. It is a 2 position header with pins on 0.100 inch centers and is compatible with AMP #640441-2 2 position IDC connector. When the interface board is installed in the supplied enclosure, P4 is connected to the front panel toggle switch. When the AVT-717 is delivered as an OEM module P4 has a jumper installed.

6.3.9 Connector P5

P5 is normally not installed. It is the background debug mode connector for the MC68332 microcontroller.

6.4 AVT-717 Enclosure Information

The following sections contain information about the AVT-717 UBP / CAN Interface board mounted in the provided enclosure. The AVT-717 board is available mounted in a rugged polycarbonate enclosure. Also included in the enclosure are two internal cable assemblies and a locking toggle switch for unit power.

6.4.1 Enclosure connector P1

Connector P1 on the AVT-717 enclosure is a DA-15P connector and will mate to any industry standard DA-15S connector. It is connected to the subject vehicle OBD-II connector through the supplied cable assembly. Only the signal lines that the AVT-717 UBP / CAN Interface uses are indicated here. In accordance with the standard there may be other signals on the OBD-II connector in the vehicle. The AVT-717 does not connect to or utilize signals or lines other than those indicated here. All other pins on this connector are 'not connected.'

Pin #	Signal
1	UBP bus.
4	Ground (chassis)
5	Ground (signal)
6	CAN - H bus.
9	SWC CAN bus.
13	External power (+12v nominal).
14	CAN - L bus.

6.4.2 Enclosure connector P2

Connector P2 on the AVT-717 enclosure is a DE-9S connector and will mate to any industry standard DE-9P connector. Depending on the configuration of the AVT-717 interface, this connector is either an RS-232 or an RS-422 communications port. Through this connection the control computer communicates with the AVT-717 UBP / CAN Interface and, hence, the subject vehicle. Hardware handshaking (RTS/CTS) is required for proper operation of the AVT-717 UBP / CAN Interface.

All signals and directions indicated are relative to the AVT-717 interface.

RS-232 Configuration

Pin Number	Name	Function / Description	
1	CD	+5v	output
2	TXD	Transmit Data	output
3	RXD	Receive Data	input
4	DTR	connected to DSR, pin #6	
5	Sig. Gnd.	Isolated from vehicle ground	
6	DSR	connected to DTR, pin #4	
7	CTS	Clear To Send	input
8	RTS	Ready To Send	output
9	RI	not used	

RS-422 Configuration

Pin Number	Name	Function / Description	
1	Sig. Gnd.	Isolated from vehicle ground	
2	CTS +	Clear To Send	input
3	CTS -	Clear To Send (inverted)	input
4	RXD +	Receive data	input
5	RXD -	Receive data (inverted)	input
6	RTS -	Ready To Send (inverted)	output
7	RTS +	Ready To Send	output
8	TXD +	Transmit data	output
9	TXD -	Transmit data (inverted)	output

7. Company Overview

Advanced Vehicle Technologies, Inc. is dedicated to providing affordable hardware, software, and technical support to the developers and users of vehicle based networks.

AVT, Inc. also offers other vehicle network products including:

- AVT-418 Multiple Interface; Ethernet connection to host.
- AVT-718 Multiple Interface; RS-232/422 connection to host.
- AVT-841 Low Cost Multiple Interface; RS-232 connection to host.
- AVT-842 Low Cost Multiple Interface; USB connection to host.
- AVT-843 Low Cost Multiple Interface; Ethernet connection to host.
- AVT-716 Multiple Interface; RS-232/422 connection to host.

Contact the factory for information on these products and our latest offerings.

The engineering staff at AVT, Inc. is experienced with multiplex bus standards including: CAN with multiple physical layers; J1850 VPW; J1850 PWM; ISO-9141; ISO-9141-2; Keyword Protocol 2000; ISO 14230; ALDL; CCD; UBP; and more.

AVT engineering staff are available to provide dedicated engineering support for a custom 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)

Home Page:	www.AVT-HQ.com
e-mail:	support@AVT-HQ.com

Revision Record:

A1: Original release.
August 1999
A2: Added the block diagram on page 9.
October 2006
A3: Updated baud rate selection tables for different firmware versions.
Added short paragraph notes for new protocol support:
UBP, LIN, GM +12v ALDL or UART.
February 2007.