ADVANCED VEHICLE TECHNOLOGIES, Inc.

AVT - 822

J1939 / J1708 Controller

Interface Control Document

and

Related Technical Information

Hardware revision “A3”
Firmware Version “1.6 (01)”
3 November 2014

1509 Manor View Road, Davidsonville, MD 21035 USA
+1-410-798-4038 (voice) support@AVT-HQ.com
www.AVT-HQ.com

Table of Contents

1. Introduction

This document describes the AVT-822 hardware and firmware.

I try to follow these rules and conventions.

NOTE:

I tried to prefix all HEX numbers with the "$" sign.
All other numbers are decimal.
If you have a question about any numbers, ask me.

NOTE:

In keeping with the SAE J1939 documentation …

Bytes in a CAN frame are numbered 1 to 8.
Bits are numbered 1 to 8.

1.1 Hardware

The AVT-822 board revision stands at "A3" – Production.

Refer to our web site for the most up-to-date information about the hardware status of each board.

Hardware status: www.AVT-HQ.com/822_hw.htm

1.2 Firmware

The firmware version stands at "1.6 (01)".

Refer to our web site for the most up-to-date information about AVT-822 firmware versions:

www.AVT-HQ.com/822_asm.htm

At present, there is no J1708 support in the firmware. Activity on the J1708 port is ignored.

1.2.1 Determining Firmware Version

Perform the following to determine the version of firmware in your unit.

- Install the host computer debug port adapter.
- Connect to a host computer running the Hex Terminal or equivalent.
- Power on the AVT-822 controller.
- The power-on notification is:

  $ 91 $ 12 indicates idle mode operation.
  $ 92 $ 04 $ xx where "xx" is the firmware version.
  example: "xx" = 16 is version 1.6.

- At any time, send the B0' command.
- The response will be: $ 92 $ 04 $ xx where "xx" is the firmware version.
- At any time, send the $ B1 $ 01 command.
- The response will be: $ 93 $ 04 $ xx $ yy
  example: "xx yy" = 16 01 is version 1.6 (01).

1.3 Commands and Responses

A list of commands, responses, error codes, notes, etc. is provided at the end of this document.

2. Glossary

Common terms, abbreviations, acronyms, and more.

$ sign	Indicates a hex number.
CAN	Controller Area Network
CAN0	CAN channel 0
CAN4	CAN channel 4
EEPROM	Electrically Erasable Programmable Read Only Memory. Usually with the form of small rows and sectors. Erase and program operations are usually done for one sector at a time.
FLASH	A form of EEPROM. Usually with the form of large rows and sectors. Erase and program operations are usually done for one sector at a time.
ISO 11898	An ISO specification for 2-wire CAN physical layer.
msec.	Milliseconds, 1/1000 of a second.

3. Electrical Interface Description

All electrical connections to the AVT-822 board are described here.

There are two (2) thru-hole solder connections on the board for each signal described here.

3.1 P2 – Power

Square pad indicates pin #1.

Pin # 1:	+8 to +36 VDC ~ 60 milliamps
Pin # 2:	Ground or supply return.

3.2 P4 – J1939 Network (CAN bus)

Square pad indicates pin #1.

Pin # 1:	J1939 CAN_H signal.
Pin # 2:	ground
Pin # 3:	J1939 CAN_L signal.

Note: The AVT-822 does NOT (normally) terminate the J1939 bus.

3.3 P7 – Outputs

Square pad indicates pin #1.

Pin # 1:
- signal name: RPM_FREQ_OUT
- type: output
- description: +5 V HCMOS compatible
  square wave, 50% duty cycle
- range: 0 or 0.2 to 3500 Hz

Pin # 2:
- signal name: VSS_FREQ_OUT
- description: +5 V HCMOS compatible
  square wave, 50% duty cycle
- range: 0 or 0.02 to 240 Hz

Pin # 3:
- signal name: REGEN_ACTIVE
- description: +5 V HCMOS compatible
  active high

Pin # 4:
- signal name: Status_02 -- not used
- type: output
- description: +5 V HCMOS compatible

Pin # 5:
- signal name: GROUND
- type: Reference ground for signals on pins 1 through 4.

3.4 P6 – Inputs

Square pad indicates pin #1.

Pin # 1:
- signal name: HIGH_IDLE_ENABLE
  +5 V HCMOS compatible.
  Passively pulled high through 10 Kohm resistor.
  Active low.
- description: When high or disconnected (inactive), the AVT-822 does NOT transmit the J1939 message to control engine idle speed.
  When pulled low (active), the J1939 message for High Idle is enabled.

Pin # 2:
- signal name: HIGH_IDLE_INCREASE
  +5 V HCMOS compatible.
  Passively pulled high through 10 Kohm resistor.
  Active low.
- description: When high or disconnected (inactive), there is no change in commanded engine idle speed.
  When pulled low (active), the commanded High Idle speed will increase 6 RPM for every 100 msec. it is held low.
  The HIGH_IDLE_ENABLE input must also be active.

Pin # 3:

signal name: SPARE_2 – not used
+5 V HCMOS compatible.
Passively pulled high through 10 Kohm resistor.

Pin # 4:

signal name: SPARE_1 – not used
+5 V HCMOS compatible.
Passively pulled high through 10 Kohm resistor.

Pin # 5:

signal name: GROUND
type: Reference ground for signals on pins 1 through 4.

3.5 P5 – J1708 Bus

Square pad indicates pin #1.

Pin # 1: J1708 A signal

Pin # 2: ground

Pin #3: J1708 B signal

The AVT-822 implements the J1708 transceiver as described in J1708 rev. OCT93.

3.6 P1 – Debug Connector

Normally NOT USED.

A separate communications adapter board can be connected to P1 which then provides an RS-232 communications port to a computer for debug and operational monitoring purposes. Refer to Section 9 for details about the debug port.

3.7 P3 – Microcontroller BDM Connector

DO NOT USE.

For factory use only. Do not connect anything to P3.

4. J1939 Message Descriptions

All J1939 messages to and from the AVT-822 board are described here.

J1939 Byte order:

Byte ‘data1’ is the first byte in the data field of a single frame, after the ID field.
Byte ‘data8’ is the last byte in the data field of a single frame.

Message ID notes:

“x” = 1 bit
“y” = 4 bits

4.1 Vehicle Speed – Reported

The AVT-822 'listens' for (receives) this message and generates the indicated output from the data read from this message.

If this message is not received within 5 seconds the output defaults to the minimum indicated value of 0.02 Hz.

Msg ID:	$ xyFEF1yy
PGN:	65265
SPN:	84
Bytes:	2:3
order:	Intel
units:	km/hr
increment:	1/256 km / hr / bit
offset:	0
output:	connector P7, pin #2
	0 or 0.02 to 240 Hz

Mathematical relationships:

Output frequency (Hz) = (((240 – 0.02) / 10) * speed) + 0.02
Where 'speed' is in miles per hour.
Output frequency is limited to 240 Hz.
J1939 vehicle speed is converted to miles per hour using this conversion
speed (mph) = (J1939 value / 412) the answer is rounded up or down
to the nearest integer.

4.2 Engine Speed – Reported

The AVT-822 'listens' for (receives) this message and generates the indicated output from the data read from this message.

If this message is not received within 5 seconds the output defaults to the minimum indicated value of 0.2 Hz.

Msg ID:	$ xyF004yy
PGN:	61444
SPN:	190
Bytes:	4:5
order:	Intel
units:	RPM
increment:	0.125 RPM / bit
offset:	0
output:	connector P7, pin #1
	0 or 0.2 to 3500 Hz

Mathematical relationships:

Output frequency (Hz) = (((3500 – 0.2) / 1600) * RPM) + 0.2
Where 'RPM' is Revolutions Per Minute.
Output frequency is limited to 1600 Hz.
J1939 engine speed is converted to RPM using this conversion

RPM = (J1939 engine speed value / 8) the answer is rounded up or down to the nearest integer.

4.3 Regen Requested – Reported

The AVT-822 'listens' for (receives) this message and generates the indicated output from the data read from this message.

Diesel Particulate Filter Lamp Command

Msg ID:	$ xyFD7Cyy
PGN:	64892
SPN:	3697
Byte:	1
bits:	1:3
output:	connector P7, pin #3 goes high if either this or the next condition is met

Exhaust System High Temperature Lamp Command

Msg ID:	$ xyFD7Cyy (64892)
PGN:	64892
SPN:	3698
Byte:	7
bits:	3:5
output:	connector P7, pin #3 goes high if either this or the previous condition is met

4.4 High Idle – Commanded

The AVT-822 transmits for this message based on the condition of the two inputs described in Section 3.4.

Msg ID:	$ xy0000yy
PGN:	0
SPN:	898
Bytes:	2:3
order:	Intel
units:	RPM
increment:	0.125 RPM / bit
offset:	0

Refer to Section 8.1 for a detailed description of the operations of this function, command, and message.

5. J1708 Message Descriptions

All J1708 messages to and from the AVT-822 board are described here.

5.1 xxx – Reported

No J1708 support at this time.

5.2 xxx – Commanded

No J1708 support at this time.

6. Operations

The AVT-822 does not have a power switch. The controller powers up and begins operations as soon as power is applied.

On power-up, the controller completes an internal initialization sequence.

Then, it executes all stored commands. See following Sections for more information.

Then it waits for commands from the host computer (if connected).

6.1 Start Up

A series of start-up commands are stored in non-volatile memory of the AVT-822. The stored commands are executed every time the AVT-822 starts up from power-on or a commanded reset.

The start-up commands set-up various operational parameters of the AVT-822. This allows the AVT-822 to be 'customized' for specific applications while "in the field".

6.2 Start Up Commands

As of the time of writing, the following start-up commands are stored during factory testing.

; switch to CAN mode
E1 99

; enable normal operations
73 11 00 01

The internal initialization routines set the CAN bus baud rate to 250 k baud.

6.3 Controller Operations

Following power-on or reset, the AVT-822 completes an internal initialization sequence and then executes the stored commands, as listed in Section 6.2.

After entering CAN mode and enabling channel CAN0, the AVT-822 executes a 14 second delay before commencing "normal" operations.

In "normal" operations the AVT-822 monitors the J1939 network for messages of interest. Refer to Section 7 for Passive Operations.

The AVT-822 also monitors the two "High Idle" inputs. Refer to Section 8 for Active Operations.

NOTE:
Only when the two inputs to the AVT-822 board are connected and forced active does the AVT-822 send messages onto the J1939 bus and thus 'Active' operations.

6.4 Operational Parameters

Some operational parameters can be stored in non-volatile memory. If not defined, they default to a specific value. The available parameters and default values are listed here.

Refer to Section 11 for details about each command to query for or change the stored value.

Missing Message Timer: The amount of time before an engine RPM message (SPN 190) or vehicle speed message (SPN 84) is declared missing and the associated output reverts to the default value.
default value: 5 seconds
command: 72 36 xx

Change High Idle delta: The RPM value the commanded engine RPM is changed per 100 msec.
default value: 6 RPM
command: 52 13 xx

TSC1 Source Address: The source address used in the TSC1 message for the High Idle function.
default value: $FF (binary: all ones)
command: 52 14 xx

Safe Minimum RPM: The minimum RPM value that would ever be transmitted in a TSC1 message during High Idle operations.
default value: 100 RPM
command: 53 11 xx yy

Safe Maximum RPM: The maximum RPM value that would ever be transmitted in a TSC1 message during High Idle operations.
default value: 2500 RPM
command: 53 12 xx yy

Host communications baud rate: Used for debug operations only.
default value: 57.6 kbaud
command: 52 67 xx

6.5 Factory Stored Parameters

The following parameters are stored during factory testing.

Missing Message Watchdog
set to 5 seconds
command: 72 36 1E

6.6 Heartbeat Indicator

A red LED on the board blinks at a nominal rate of 3 times per second to indicate proper operation of the microcontroller.

7. Passive Operations

After power-on, there is a 14 second delay. Then, the AVT-822 starts monitoring the J1939 network for messages and SPNs of interest.

NOTE:
When the AVT-822 is operating passively it does not transmit any messages to the J1939 bus.

The AVT-822 monitors the J1939 bus for the SPNs listed below. The information obtained is used to generate the outputs described.

Refer also to Section 4 for related information.

7.1 SPN # 190

This is the J1939 reported engine speed. This number is converted to RPM. That value is then used to compute the frequency of a square wave output signal (50% duty cycle). The output signal defaults to 0.2 Hz if this SPN is not received from the J1939 bus. The output frequency is updated every time the SPN is received from the J1939 bus. If the SPN goes missing from the J1939 bus for more than 5 seconds (that time limit can be changed), then this output returns to the default value of 0.2 Hz.

7.2 SPN # 84

This is the J1939 reported vehicle speed. This number is converted to miles per hour. That value is then used to compute the frequency of a square wave output signal (50% duty cycle). The output signal defaults to 0.02 Hz if this SPN is not received from the J1939 bus. The output frequency is updated every time the SPN is received from the J1939 bus. If the SPN goes missing from the J1939 bus for more than 5 seconds (that time limit can be changed), then this output returns to the default value of 0.02 Hz.

7.3 SPN # 3697 and SPN # 3698

This J1939 data commands the Diesel Particulate Filter Lamp to be lit (which indicates to the operator that a Regen is required). If either of these SPNs go active (lamp lit) – then REGEN requested output (P7 pin #3) goes active (low). The output signal will stay active (LOW) until the REGEN timer (10 minutes) expires or both SPNs go inactive – whichever is later.

8. Active Operations

8.1 High Idle Operations

When commanded to do so, the AVT-822 will transmit a message to the J1939 bus to command a change to the engine RPM.

NOTE:
The device that controls the two “High Idle” inputs to the AVT-822 assumes all risk and responsibility for any and all actions and consequences that result.

When the HIGH_IDLE_ENABLE input is in the passive state (high) then the High Idle function is disabled and the AVT-822 does not transmit any messages to the J1939 bus.

The HIGH_IDLE_INCREASE input is ignored at all times when the HIGH_IDLE_ENABLE input is disabled (high).

When the HIGH_IDLE_ENABLE input is first pulled low to the active state, there is no change in the operation of the AVT-822. At this point the High Idle function begins monitoring the state of the HIGH_IDLE_INCREASE input.

When the HIGH_IDLE_INCREASE input is pulled low to the active state the AVT-822 starts a timer.

After the HIGH_IDLE_INCREASE input has been held low (active) for 100 msec. the AVT-822 takes the last five reported engine RPM values and computes the average. The AVT-822 then adds 6 RPM to the average to determine the new RPM value to command the engine to go to. At this point the AVT-822 constructs a J1939 TSC1 message with the new engine RPM encoded. The AVT-822 begins transmitting the TSC1 message every 100 msec.

The AVT-822 continues to monitor the state of the HIGH_IDLE_INCREASE input. For every full 100 msec interval that the HIGH_IDLE_INCREASE input is held low (active) the AVT-822 will increase the commanded RPM by 6 RPM and update the TSC1 message it is transmitting.

If the HIGH_IDLE_INCREASE input is held low for any interval less than 100 msec. – it is ignored. For example, if the HIGH_IDLE_INCREASE input is held low for 280 msec. After 100 msec. the commanded engine RPM will be increased by 6 RPM. After 200 msec. the commanded engine RPM will increase again by 6 RPM (12 RPM total, so far). After 280 msec. the external device releases the HIGH_IDLE_INCREASE input. The last interval is 80 msec., which is less than 100 msec.; so it is ignored.

When the HIGH_IDLE_ENABLE input is released the AVT-822 immediately starts decreasing the commanded RPM down by 6 RPM per every 100 msec. This continues until the AVT-822 gets the commanded RPM back down to the original RPM value it noted when the process started. At that point the TSC1 messages are changed to release the engine controller from external control and the AVT-822 stops transmitting the TSC1 message. This ends AVT-822 control of the engine RPM.

Note that if the HIGH_IDLE_ENABLE input is toggled high (disable) and then low again (enable) that sequence completely resets the High Idle function.

8.2 TSC1 Message Construction

Here is a detailed listing of the construction of the TSC1 message that the AVT-822 transmits to control engine RPM when the High Idle function is enabled and operating.

ID bits 28:08 are set to 0 (binary zero).

ID bits 07:00 are the source address as read from non-volatile memory. If a source address is not stored in non-volatile memory, then the source address be all 1 (binary one).

Data byte 1: in binary: 1111 0100.
bits 8:7 are 11 = not defined.
bits 6:5 are SPN 897, set to binary 11, Override Control Mode Priority to Low.
bits 4:3 are SPN 696, set to binary 01, Engine Requested Speed Control to
Stability Optimized for Driveline Disengaged and Non-Lockup Conditions.
bits 2:1 are SPN 695, set to binary 01, Engine Override Control Mode to Speed Control - Govern speed to the included “desired speed” value.

Data bytes 2 and 3 hold the commanded engine RPM.
This field will be updated when the High Idle function is enabled and operating.

Data byte 4: in binary: 0000 0000.
SPN 518; torque limit set to zero.
Since this message specifies speed control mode, this value should be ignored.

Data byte 5: in binary: 0001 0100.
bits 8:4 are SPN 3350, TSC1 Control Purpose, set to binary 00010 which means purpose P3, PTO Governor.
bits 3:1 are SPN 3349, TSC1 Transmission Rate, set to binary 100 which means 100 msec.

Data byte 6: in binary: 1111 1000
bits 8:5 are undefined and set to binary one.
bits 4:1 are SPN 4191, requested torque, set to binary 1000 which means not available.
Since this message specifies speed control mode, this value should be ignored.

Data byte 7: in binary: 1111 1111
all bits are undefined.

Data byte 8: in binary: xxxx xxxx
bits 8:5 is a rolling counter.
bits 4:1 is a message checksum.
Each time the AVT-822 prepares to transmit this message, the counter is updated and then the checksum is computed.

9. Debug Port

The AVT-822 uses an RS-232 serial communications connection to the host computer for debug and monitor functions.

9.1 Connection to Debug Computer

A special debug adapter is required.

The debug adapter must be plugged onto connector P1 of the AVT-822 board.

Be sure to align Pin #1 of the debug adapter to Pin #1 of P1.

Then, connect the debug adapter to the host computer …

Directly, if the host computer has an RS-232 port.
Or
Use a USB to RS-232 adapter.

The AVT-822 uses RTS/CTS hardware handshaking for flow control. The host computer must be configured for RTS/CTS hardware handshaking to proper operations.

Do NOT use software flow control (XON/XOFF).

For reference, the signals of AVT-822 connector P1 are listed here.
(These are TTL level signals.)

Pin #	Signal	Direction
1	+5 vdc	AVT-822 output
2	Ground	
3	TX data	AVT-822 output
4	RX data	AVT-822 input
5	RTS	AVT-822 output
6	CTS	AVT-822 input

AVT-822 Debug Connector P1 Signals

For reference, the signals of the debug adapter DE-9S connector pins are listed here.

Pins #1, #6, and #9 are tied together on the AVT-822 board through 10 ohm resistors.

Pin #	Signal	Direction
1		No connection
2	TX data	Adapter output
3	RX data	Adapter input
4		
5	GND	
6		
7	CTS	Adapter input
8	RTS	Adapter output
9		

AVT-822 Debug Adapter DE-9S Connector Signals

9.2 Baud Rate

The serial communications baud rate between the AVT-822 and the host computer is 57.6 kbaud (factory default).

9.3 Hex Terminal

All debug communications between the AVT-822 and the host computer is in binary format. Therefore, you can not use Hyperterm, TerraTerm or related terminal applications.

The AVT "Hex Terminal" application is available from the AVT web site. It is free.

http://www.AVT-HQ.com/download.htm#Hex_Term_SW

10. Idle Mode - Commands

B: Firmware version.

B0:	Request firmware version number.
B1 01:	Request extended firmware version number.

D: Operational mode.

D0:	Request operational mode report.

E: Mode switch.

E1 94:	Reserved (do not use).
E1 96:	Reserved (do not use).
E1 99:	Switch to CAN mode.
E1 DD:	Switch to J1708 mode (do not use).

F: Model Query and Reset

F0:	Query for model number.
F1 A5:	Restart the AVT-84x (a form of software reset).

10.1 Idle Mode - Responses

2: Error reports.

22 34 xx: Command time-out.
xx: header byte of offending command.

---

22 77 xx: Switch mode error. "xx" = specific error byte.

01:	start address equals $0000.
02:	start address equals $FFFF.
03:	start address less than or equal to $8000.
04:	start address equal to or greater than $BFFF.
05:	expected checksum equals $0000.
06:	expected checksum equals $FFFF.
07:	byte count to sum = $0000.
08:	checksums are not equal.

3: Invalid command.

31 xx: Invalid command.
xx: header byte of offending command.

10.2 Other Responses

2: Error reports.

21 70: Backdoor is disabled.

21 71: Backdoor access attempt failed.

21 84: Command buffer mode fault.

11. CAN Mode - Commands

High nibble, bits b7 - b4: Command type.

0: Packet for message to be transmitted to the network.

Format "0x"

0p xy tt vv ww zz mm nn ... :

- p: count of bytes to follow.
- x:
  - b7: IDE.
    - 0: 11-bit ID.
    - 1: 29-bit ID.
  - b6: RTR.
    - 0: normal frame.
    - 1: RTR true, remote transmit request.
  - b5: 0
  - b4: 0
- y: Channel number: 0, 4, 5, 6.
- tt vv: 11-bit ID, right justified.
- tt vv ww zz: 29-bit ID, right justified.
- mm nn ...: data.

Maximum of 8 data bytes.

2: Reset.

- 21 04: Reset FIFO #1.
- 21 05: Reset FIFO #2.
- 21 0A: Reset CAN0.
- 21 0B: Reset CAN4.

3:

4:

5: Configuration.

- 51 09: Message count reports status query.
- 52 09 00: Disable message count reports. [Default]
- 52 09 01: Enable message count reports.

- 51 11: Query for Stored Safe High Idle Minimum RPM.
- 53 11 xx yy: Set Stored Safe High Idle Minimum RPM.
  Value is RPM, integer, word, high byte, low address.

---

51 12: Query for stored Safe High Idle Maximum RPM.
53 12 xx yy: Set stored Safe High Idle Maximum RPM.
Value is RPM, integer, word, high byte, low address.

---

51 13: Query for stored Change Idle RPM Delta.
52 13 xx: Set stored Change Idle RPM Delta.
Value is RPM, integer, byte, per 100 msec.

---

51 14: Query for stored TSC1 source address.
52 14 xx: Set stored TSC1 source address.

---

51 40: Transmit acks query.
52 40 00: Do not send transmit acks to host.
52 40 01: Send transmit acks to host. [Default]

---

52 4C xx: Command processing delay.
Delay is 'xx' timer ticks (5x 63 command).
Only useful between commands; does not otherwise affect operations.

---

53 5E DE AD: Mass erase all EEPROM memory (User Space).
(This includes user space, auto start commands, baud byte, etc.)
(Not a secure command.)

---

51 60: Query for lock/unlock status (User Space).
52 60 00: Lock the EEPROM.
55 60 rr ss tt vv: Request to unlock the EEPROM.
rr ss tt vv = security bytes.
(Not a secure command.)

---

54 61 xx yy rr: Read EEPROM memory (User Space).
xx yy = EEPROM memory location.
rr = count of bytes to read. Maximum of $0B.
(Secure command.)

---

57 62 xx yy rr ss tt vv: Write EEPROM memory (User Space).
xx yy = EEPROM memory location.
rr ss tt vv: bytes to be written.

(Secure command.)

---

51 63: Master timer status query.
52 63 xx: Master timer setting.

xx:		
	01	98.30 msec. [Default]
	02	49.15 msec.
	03	20.48 msec.
	04	10.24 msec.
	05	5.12 msec.

---

51 67: Query for host baud rate setting.
52 67 01: Set host baud rate to 19.2 kbaud.
52 67 02: Set host baud rate to 38.4 kbaud.
52 67 03: Set host baud rate to 57.6 kbaud.
52 67 04: Set host baud rate to 115.2 kbaud.
Note: New setting does not take affect until unit is reset; either power-on reset or software reset (F1 A5).

---

51 68: Query for the 4 security bytes.
55 68 rr ss tt vv: Set the 4 security bytes.
(Secure command.)

---

51 6A: Query for red LED blink rate.
52 6A xx: Set red LED blink rate.
00 = red LED off.
xx = red LED blink rate; interval is 174.8 msec.
FF = red LED on.

6:

7: CAN configuration.

71 0A: Request baud rate settings for all CAN channels.
72 0A 0x: Request baud rate setting for CAN channel.
x: CAN channel, 0 or 4.
73 0A 0x yy: Set baud rate for CAN channel.
x: CAN channel number: 0 or 4.
yy:
00: User specified using 74 0B 0x rr ss command.
01: 1 Mbps.
02: 500 Kbps. [Default for CAN0.]
03: 250 Kbps.
04: 125 Kbps.

0A: 33.333 Kbps. [Default for CAN4].
0B: 83.333 Kbps.

---

71 0B: Request Bit Timing Register (BTR) settings for all CAN channels.

72 0B 0x: Request BTR settings for CAN channel.
- x: CAN channel number: 0 or 4.

74 0B 0x rr ss: Set Bit Timing Registers (BTR) for CAN channel.
- x: CAN channel, 0 or 4.
- rr: Bit Timing Register 0 setting.
- ss: Bit Timing Register 1 setting.

---

71 11: Request operational mode status for all CAN channels.

72 11 0x: Request operational mode status for CAN channel.
- x: CAN channel, 0 or 4.

73 11 0x 0y: Set operational mode for CAN channel.
- x: CAN channel, 0 or 4.
- y:
  - 0: Disabled. [Default for CAN0 and CAN4.]
  - 1: Enabled for normal operations.
  - 2: Enabled for listen only operations.

NOTE: The periodic message setup command (7x 18) has the CAN channel and message number fields reversed as compared to all other commands.

---

73 18 vv 0y: Periodic message setup query.
- vv: Message number, $01 to $10.
- y: Channel number: 0, 4, 5, 6.

7x 18 vv xy tt vv ww zz mm nn ... Periodic message setup command.
- vv: Message number, $01 to $20.
- x:
  - b7: IDE.
    - 0: 11-bit ID.
    - 1: 29-bit ID.
  - b6: RTR.
    - 0: normal frame.
    - 1: RTR true, remote transmit request.
  - b5: 0
  - b4: 0
- y: Channel number: 0, 4, 5, 6.
- tt vv: 11-bit ID, right justified.
- tt vv ww zz: 29-bit ID, right justified.
- mm nn ...: data field.

---

73 1A 0y zz: Periodic message disable/enable status query.
- y: Channel number: 0, 4, 5, 6.
- zz: Message number, $01 to $20.

74 1A 0y zz 0v: Periodic message disable/enable command.
- y: CAN channel: either '0' or '4'
- zz: Message number, $01 to $20.
- v: 0 disabled.
  1 'normal' mode enabled.
  2 'slave' mode enabled.
  3 both modes enabled.

---

73 1B 0x yy: Periodic message interval count status query.
- x: Channel number: 0, 4, 5, 6.
- yy: Message number, $01 to $20.

74 1B 0x yy vv: Periodic message interval count command.
- x: Channel number: 0, 4, 5, 6.
- yy: Message number, $01 to $20.
- vv: interval count.

---

72 1C 0x: Disable all periodic messages of one CAN channel.
- x: Channel number: 0, 4, 5, 6.

72 1C EE: Disable all periodic messages, all CAN channels.
(Note: the setup for each periodic message is not affected.)

---

71 2A: Report all acceptance IDs for all CAN channels.

72 2A 0x: Report all acceptance IDs for CAN channel.
- x: CAN channel, 0 or 4.

73 2A 0x 0z: Report specified acceptance ID.
- x: CAN channel, 0 or 4.
- z: Acceptance ID number; from 00 on up.
  Number depends on ID/Mask mode.

7x 2A xy 0z rr ss tt vv: Set acceptance ID.
- x: b7: IDE.
  0: 11-bit ID.
  1: 29-bit ID.
  b6: RTR.
  0: normal frame.
  1: RTR true, remote transmit request.
  b5: 0
  b4: 0
- y: CAN channel, 0 or 4.
- z: Acceptance ID number; from 00 on up.

Number depends on ID/Mask mode.
rr: Acceptance ID value when ID/Mask mode = 8.
rr ss: Acceptance ID value when ID/Mask mode = 4.
rr ss: Acceptance ID value when ID/Mask mode = 2
and IDE = 0 (11-bit).
rr ss tt vv: Acceptance ID value when ID/Mask mode = 2
and IDE = 1 (29-bit).

---

71 2B: Report ID/Mask mode for all CAN channels.
72 2B 0x: Report ID/Mask mode for CAN channel.
x: CAN channel, 0 or 4.
73 2B 0x 0y: Set ID/Mask mode for CAN channel.
x: CAN channel, 0 or 4.
y: 2: Two 32-bit IDs and masks.
4: Four 16-bit IDs and masks.
8: Eight 8-bit IDs and masks.

---

71 2C: Report all masks for all CAN channels.
72 2C 0x: Report all masks for CAN channel.
x: CAN channel, 0 or 4.
73 2C 0x 0z: Report specified mask.
x: CAN channel, 0 or 4.
z: Mask number; from 00 on up.
Number depends on ID/Mask mode.
7x 2C xy 0z rr ss tt vv: Set mask.
x: b7: IDE bit.
0: must match.
1: don't care.
b6: RTR bit.
0: must match.
1: don't care.
b5: 0
b4: 0
y: CAN channel, 0 or 4.
z: Mask number; from 00 on up.
Number depends on ID/Mask mode.
rr: Mask value when ID/Mask mode = 8.
rr ss: Mask value when ID/Mask mode = 4.
rr ss: Mask value when ID/Mask mode = 2
and acceptance ID has IDE = 0 (11-bit).
rr ss tt vv: Mask value when ID/Mask mode = 2.
acceptance ID has IDE = 1 (29-bit).

---

73 2D 0x 0y: Acceptance ID register direct read.
- x: CAN channel, 0 or 4.
- y: Acceptance ID register number, 0 to 7.

74 2D 0x 0y zz: Acceptance ID register direct write.
- x: CAN channel, 0 or 4.
- y: Acceptance ID register number, 0 to 7.
- zz: Register value to write.

---

73 2E 0x 0y: Mask register direct read.
- x: CAN channel, 0 or 4.
- y: Mask register number, 0 to 7.

74 2E 0x 0y zz: Mask register direct write.
- x: CAN channel, 0 or 4.
- y: Mask register number, 0 to 7.
- zz: Register value to write.

---

71 36: Query for stored Missing Message Timer.

72 36 xx: Set stored Missing Message Timer.
xx increments of 167.8 msec.

8:

9:

A:

B: Firmware version.

B0: Request firmware version number.

B1 01: Request extended firmware version number.

C:

D: Operational mode.

D0: Request operational mode report.

E: Mode switch.

E1 94: Reserved (do not use).

E1 96: Reserved (do not use).

E1 99: Switch to CAN mode.

E1 DD: Switch to J1708 mode (do not use).

F: Model Query and Reset

F0: Query for model number.

F1 A5: Restart the AVT-84x (a form of software reset).

11.1 CAN Mode - Responses

High nibble, bits b7 - b4: Command type.

0: Packet for message received from the network.

0p xy tt vv ww zz mm nn ... :

p: count of bytes to follow.

x: b7: IDE.
- 0: 11-bit ID.
- 1: 29-bit ID.

b6: RTR.
- 0: normal frame.
- 1: RTR true, remote transmit request.

b5: 0

b4: 0

y: Channel number: 0, 4, 5, 6.

tt vv: 11-bit ID, right justified.

tt vv ww zz: 29-bit ID, right justified.

mm nn ...: data field.

02 0y 0z: Transmit ack.
- y: Channel number: 0, 4, 5, 6.
- z: Transmit buffer number.

2: Error reports.

21 0E: Transmit command too long.

---

21 11: High Idle overflow; PM#00 manager.

---

21 12: High Idle underflow; PM#00 manager.

---

21 13: Safety minimum RPM; PM#00 manager.

---

21 14: Safety maximum RPM; PM#00 manager.

---

21 15: TSC1 safety disabled; PM#00 manager.

---

21 16: RPM watchdog failed with High Idle enabled.

---

21 17: High Idle state error, pin low.

---

21 18: High Idle state error, pin high.

---

21 19: Change Idle state error, pin low.

---

21 1A: Change Idle state error, pin high.

---

21 1B: Change Idle, RPM too high.

---

22 2C xx: Serial comms with host error.

xx:

- b7: transmit data register empty.
- b6: transmit complete.
- b5: receive data register full.
- b4: idle.
- b3: overrun.
- b2: noise flag.
- b1: framing error.
- b0: parity fault.

---

22 34 xx: Command time-out.
xx: header byte of offending command.
[0.5 seconds.]

---

21 70: FLASH error.

---

21 71: FLASH error.

---

21 72: reFLASH routine: erase row; or
EEPROM routines: mass erase or erase sector or program sector

CBEIF and CCIF not set.

---

22 73 xx: reFLASH routine; erase row error detected; FSTAT follows.

---

23 74 xx yy: reFLASH routine; blank check failed
EEPROM routine; mass erase blank check failed; xx yy = error count
EEPROM routine; erase sector; blank check failed.

---

21 75: reFLASH routine; program row; CBEIF and CCIF not set.

---

21 76: reFLASH routine; program row; error detected; FSTAT follows.

---

22 77 xx: Switch mode error. "xx" = specific error byte.
- 01: start address equals $0000.
- 02: start address equals $FFFF.
- 03: start address less than or equal to $8000.
- 04: start address equal to or greater than $BFFF.
- 05: expected checksum equals $0000.
- 06: expected checksum equals $FFFF.
- 07: byte count to sum = $0000.
- 08: checksums are not equal.

---

21 79: No instruction trap.

---

21 7A: COP fail reset.

---

21 7B: Clock monitor reset.

2x 7C: EEPROM protection enabled.

---

2 7D: EEPROM 'PVIOL' or 'ACCERR' bit set.

---

22 7E xx: Write EEPROM errors.
xx = 'ESTAT'

---

22 7F xx CAN processing error.

00:
01: CAN0; 0x transmit processing error.
02: CAN4; 0x transmit processing error.
03: Invalid CAN channel number.
04: CAN0; channel not configured to transmit.
05:
06: CAN0; transmit command too short, 11-bit.
07: CAN0; transmit command too long, 11-bit.
08: CAN0; transmit command too short, 29-bit.
09: CAN0; transmit command too long, 29-bit.
0A:
0B: CAN0; fail to enter sleep mode.
0C: CAN0; fail to enter init mode.
0D: CAN0; fail to exit init mode (enable listen).
0E: CAN0; fail to exit init mode (enable normal).
0F:

20:

30:

40:
41:
42:
43:
44: CAN4; channel not configured to transmit.
45:
46: CAN4; transmit command too short, 11-bit.
47: CAN4; transmit command too long, 11-bit.
48: CAN4; transmit command too short, 29-bit.
49: CAN4; transmit command too long, 29-bit.
4A:
4B: CAN4; fail to enter sleep mode.
4C: CAN4; fail to enter init mode.
4D: CAN4; fail to exit init mode (enable listen).
4E: CAN4; fail to exit init mode (enable normal).
4F:

50:
51: CAN0; mask mode not equal mask status, 7x_2B.
52: CAN4; mask mode not equal mask status, 7x_2B.
53: Mask mode not equal mask status, 7x_2B.
54: Invalid id mode in CAN_rpt_all_ids.
55: Invalid mask mode in CAN_rpt_all_masks.

56: Invalid CAN channel in 7x_2C.
57: Invalid CAN channel in 7x_2C.
58: Invalid mask number for mask mode in 7x_2C.
59: Invalid mask mode in 7x_2C.
5A: CAN0; Invalid mask number in 7x_2C.
5B: CAN4; Invalid mask number in 7x_2C.
5C: Invalid channel number in 7x_2C.
5D: Incorrect header byte in 7x_2C. (Mode 2, 11-bit.)
5E: Incorrect header byte in 7x_2C. (Mode 2, 29-bit.)
5F: Incorrect header byte in 7x_2C. (Mode 4.)

60: Incorrect header byte in 7x_2C. (Mode 8.)
61: Mask mode error in 7x_2C.
62:
63:
64:
65:
66:
67:
68:
69:
6A:
6B: Invalid CAN channel in 7x_2A.
6C: Invalid CAN channel in 7x_2A.
6D: Invalid mask number for mask mode in 7x_2A.
6E: Invalid mask mode in 7x_2A.
6F: Invalid mask number in 7x_2A.

70: Invalid mask number in 7x_2A.
71: Invalid channel number in 7x_2A.
72: Incorrect header byte in 7x_2A. (Mode 2, 11-bit.)
73: Incorrect header byte in 7x_2A. (Mode 2, 29-bit.)
74: Incorrect header byte in 7x_2A. (Mode 4.)
75: Incorrect header byte in 7x_2A. (Mode 8.)
76: Invalid mask mode in 7x_2A.
77:

---

23 81 xx yy CAN0 error report.

xx:

b7: 0
b6: 0
b5: 0
b4: DLC > 8 in 7x_18 routine; 29-bit ID.
b3: DLC > 8 in 7x_18 routine; 11-bit ID.
b2: DLC > 8 in receive manager.

b1: CAN wake up interrupt error.
b0: CAN error interrupt.

yy: Copy of CAN0 rflg register.
b7: wake up interrupt flag.
b6: CAN status change interrupt flag.
b5: receiver status bit 1.
b4: receiver status bit 0.
b3: transmitter status bit 1.
b2: transmitter status bit 0.
b1: overrun interrupt flag.
b0: receive buffer full flag.

receive status bits
00: receive ok; receive error count between 0 and 96
01: receive warning; receive error count between 97 and 127
10: receive error; receive error count greater than 127
11: bus off, transmit error count greater than 255

transmit status bits
00: transmit ok; transmit error count between 0 and 96
01: transmit warning; transmit error count between 97 and 127
10: transmit error; transmit error count greater than 127
11: bus off, transmit error count greater than 255

---

23 82 xx yy CAN4 error report.
xx:
b7: 0
b6: 0
b5: 0
b4: DLC > 8 in 7x_18 routine; 29-bit ID.
b3: DLC > 8 in 7x_18 routine; 11-bit ID.
b2: DLC > 8 in receive manager.
b1: CAN wake up interrupt error.
b0: CAN error interrupt.

yy: Copy of CAN4 rflg register.
b7: wake up interrupt flag.
b6: CAN status change interrupt flag.
b5: receiver status bit 1.
b4: receiver status bit 0.
b3: transmitter status bit 1.
b2: transmitter status bit 0.
b1: overrun interrupt flag.
b0: receive buffer full flag.

---

21 84: Command buffer mode fault.

---

22 8C xx: CAN transmit timeout.

3: Command error.

31 xx: xx = Header byte of message in error.

3:

4:

5: Configuration reports.

62 09 00: Message count reports disabled. [Default]
62 09 01: Message count reports enabled.

---

63 00 0F xx: Message count report.
xx is count of messages received in 1.0068 seconds.

---

63 11 xx yy: Stored Safe High Idle Minimum RPM.
Value is RPM, integer, word, high byte, low address.

---

63 12 xx yy: Stored Safe High Idle Maximum RPM.
Value is RPM, integer, word, high byte, low address.

---

62 13 xx: Stored Change Idle RPM Delta.
Value is RPM, integer, byte, per 100 msec.

---

52 14 xx: Stored TSC1 source address.

---

62 40 00: Transmit acks to host are disabled.
62 40 01: Transmit acks to host are enabled. [Default.]

---

62 4C xx: Command processing delay.

Delay is ‘xx’ timer ticks (5x 63 command).

---

61 5E: All EEPROM memory space erased (erased state = $FF).

---

62 60 00: EEPROM is locked. [Default.]
62 60 01: EEPROM is unlocked.

6x 61 xx yy rr ss tt ...: EEPROM memory read.
xx yy = EEPROM memory location.
rr ss tt ... bytes read.

---

67 62 xx yy rr ss tt vv: EEPROM memory written and read.
xx yy = EEPROM memory location.
rr ss tt vv: bytes read.

---

62 63 xx: Master timer setting.

xx:		
	01	98.30 msec.
	02	49.15 msec.
	03	20.48 msec.
	04	10.24 msec.
	05	5.12 msec.

---

62 67 01: Host baud rate is set for 19.2 kbaud.
62 67 02: Host baud rate is set for 38.4 kbaud.
62 67 03: Host baud rate is set for 57.6 kbaud.
62 67 04: Host baud rate is set for 115.2 kbaud.

---

65 68 rr ss tt vv: Report of the 4 security bytes.

---

62 6A xx: Red LED blink rate.
00 = red LED off.
xx = red LED blink rate; interval is 174.8 msec.
FF = red LED on.

6:

7: CAN configuration reports.

83 0A 0x yy: Baud rate for CAN channel.
- x: CAN channel, 0 or 4.
- yy:
  - 00: User specified using 74 0B 0x rr ss command.
  - 01: 1 Mbps.
  - 02: 500 Kbps. [Default for CAN0.]
  - 03: 250 Kbps.
  - 04: 125 Kbps.
  - 0A: 33.333 Kbps. [Default for CAN4].
  - 0B: 83.333 Kbps.

---

84 0B 0x rr ss: Bit Timing Registers (BTR) for CAN channel.
- x: CAN channel, 0 or 4.
- rr: Bit Timing Register 0 setting.
- ss: Bit Timing Register 1 setting.

---

83 11 0x 0y: Operational mode for CAN channel.
- x: CAN channel, 0 or 4.
- y:
  - 0: Disabled. [Default for CAN0 and CAN4.]
  - 1: Enabled for normal operations.
  - 2: Enabled for listen only operations.

NOTE: The periodic message setup command (7x 18) have the CAN channel and message number fields reversed as compared to all other commands.

8x 18 vv xy tt vv ww zz mm nn ... Periodic message setup.
- vv: Message number, $01 to $20.
- x:
  - b7: IDE.
    - 0: 11-bit ID.
    - 1: 29-bit ID.
  - b6: RTR.
    - 0: normal frame.
    - 1: RTR true, remote transmit request.
  - b5: 0
  - b4: 0
- y: CAN channel, 0 or 4.
- tt vv: 11-bit ID, right justified.
- tt vv ww zz: 29-bit ID, right justified.
- mm nn ...: data field.

---

84 1A 0x yy 0v: Periodic message disable/enable status.
- x: CAN channel, 0 or 4.

yy: Message number, $01 to $20.
v: 0 disabled.
1 'normal' mode enabled.
2 'slave' mode enabled.
3 both modes enabled.

---

84 1B 0x yy vv: Periodic message interval count.
x: CAN channel, 0 or 4.
yy: Message number, $01 to $20.
vv: interval count.

---

82 1C 0x: All periodic messages of CAN channel 'x' disabled.
82 1C EE: All periodic messages, all CAN channels, disabled.

8x 2A xy 0z rr ss tt vv: Report acceptance ID.
x: b7: IDE.
0: 11-bit ID.
1: 29-bit ID.
b6: RTR.
0: normal frame.
1: RTR true, remote transmit request.
b5: 0
b4: 0
y: CAN channel, 0 or 4.
z: Acceptance ID number.
Number depends on ID/Mask mode.
rr: Acceptance ID value when ID/Mask mode = 8.
rr ss: Acceptance ID value when ID/Mask mode = 4.
rr ss: Acceptance ID value when ID/Mask mode = 2.
and IDE = 0 (11-bit).
rr ss tt vv: Acceptance ID value when ID/Mask mode = 2
and IDE = 1 (29-bit).

---

83 2B 0x 0y: Report ID/Mask mode for CAN channel.
x: CAN channel, 0 or 4.
y: 2: Two 32-bit IDs and masks.
4: Four 16-bit IDs and masks.
8: Eight 8-bit IDs and masks.

8x 2C xy 0z rr ss tt vv: Report mask.
x: b7: IDE bit.

0: must match.
1: don't care.
b6: RTR bit.
0: must match.
1: don't care.
b5: 0
b4: 0
y: CAN channel, 0 or 4.
z: Mask number.
Number depends on ID/Mask mode.
rr: Mask value when ID/Mask mode = 8.
rr ss: Mask value when ID/Mask mode = 4.
rr ss: Mask value when ID/Mask mode = 2
and acceptance ID has IDE = 0 (11-bit).
rr ss tt vv: Mask value when ID/Mask mode = 2
and acceptance ID has IDE = 1 (29-bit).

---

84 2D 0x 0y zz: Report acceptance ID register.
x: CAN channel, 0 or 4.
y: Acceptance ID register number, 0 to 7.
zz: Register value read.

---

84 2E 0x 0y zz: Report mask register.
x: CAN channel, 0 or 4.
y: Mask register number, 0 to 7.
zz: Register value read.

---

82 36 xx: Stored Missing Message Timer.
xx increments of 167.8 msec.

8:

9: Board status information.

92 04 xx: Firmware version report. Firmware version is 'xx'.
91 0C: FIFO1 or FIFO2 reset.
91 10: CAN operations.
91 12: In Idle mode.
91 24: CAN0 reset.
91 25: CAN4 reset.
91 26: In FLASH3 mode.
91 2E: In FLASH1 mode.

A:

B:

C:

D:

E:

F:

12. J1708 – Commands

High nibble, bits b7 – b4: Command type.
Low nibble, bits b3 – b0: Count of bytes to follow.

Note:
xxx
xxx
xxx

0:

1:

2:

3:

4:

5:

xxx

---
xxx

---
xxx

6:

xxx

---
xxx

7:

8:

9:

A:

B:

C:

D:

E:

F:

12.1 J1708 – Responses

High nibble, bits b7 – b4: Response type.
Low nibble, bits b3 – b0: Count of bytes to follow.

Note:
xxx

0:

1:

2:

3:

4:

5: ____

6: ____

7: ____

8: ____

9: ____

A: ____

B: ____

C: ____

D: ____

E: ____

F: ____

13. Questions ??

Contact the factory by e-mail, phone, or fax.
Contact information is provided here and on the bottom of page 1.

Post: 1509 Manor View Road
Davidsonville, MD 21035 USA

Phone: +1-410-798-4038

E-mail: Support@AVT-HQ.com
Web site: http://www.AVT-HQ.com/