

18 August, 1999

## AVT-716 - KeyWord Protocol 2000 Mode

The AVT-716 Interface unit is capable of connecting to and communicating with several different vehicle networks (*not simultaneously*). Among them is the network known as Keyword Protocol 2000 (abbreviated here as KWP2000). This document describes some operations of the AVT-716 that are unique to KWP2000 operations.

### Introduction

Unlike most other network protocols KWP2000 operations primarily describe communications between an off-board tester (the AVT-716) and the on-board network.

Before any communications between the AVT-716 unit and the vehicle network can take place, the communications link must be established through an initialization process. (This is a logical operation, not a physical one.) The KWP2000 specification calls out three different methods of initialization and the AVT-716 supports all three.

Another important and unique aspect of operations in KWP2000 mode is that it supports messages with a data field length of 255 bytes. Special accommodations have been made in the AVT-716 firmware for KWP2000 mode operations to support these very long messages.

### Information

Detailed information on KWP2000 mode commands and responses can be obtained from the "Master Commands and Responses" document that is available for downloading (in PDF format) from our web site.

Key Word Protocol 2000 is covered under ISO specification 14230 and can be obtained from both SAE ([www.sae.org](http://www.sae.org)) or ANSI ([www.ansi.org](http://www.ansi.org)).

### Operation

The AVT-716 supports both K-line and L-line connections to the vehicle. However, only the K-line connection is required.

With the AVT-716 connected and turned on, issue the command \$E1 \$DD to switch to KWP2000 mode. The AVT-716 should respond with \$91 \$0F to confirm that it has successfully switched to KWP2000 mode and is awaiting operations.

At this point the AVT-716 is 'un-initialized' and disconnected from the bus (by an internal relay). To connect to and communicate with the vehicle there are four options:

- Conduct CARB initialization.
- Conduct 5 baud initialization.
- Conduct Fast initialization.
- Connect in 'monitor' mode.

The three initialization methods are described in the ISO specification. The AVT-716 conducts the initialization in accordance with the specification.

The 'monitor' mode permits connection to and communications with the vehicle without having to conduct any of the initializations. This is not a sanctioned mode of operation but can be useful during test and debug operations or other engineering purposes.

### Initialization

The AVT-716 supports all three initialization methods (mentioned above) and the desired method is specified by the operator through the \$61 \$xx command. In order to invoke an initialization method the current operational mode must be "un-initialized."

For both CARB and 5-baud method, successful initialization will be followed by the \$71 \$03 report meaning that the initialization was completed successfully. Following successful initialization the communication link is established and operational.

For the Fast Initialization method it is up to the host to determine if the initialization was successful. To invoke Fast Initialization the host sends the command:

\$6x \$13 \$aa \$bb \$cc ...

where:

x - byte count to follow

aa bb cc ... is the "Start Communication Service" request.

Assuming the initialization sequence follows the specification the AVT-716 will report a message from the bus in normal format. This message is the "Start Communication Service" response. The AVT-716 will then report \$71 \$03 to indicate that the initialization completed successfully.

Success, according to the 716 unit, is determined by the fact that the 'start' message went out and the response was received within the allotted time interval. The host should examine the "Start Communication Service" response for accuracy. If a negative response was received or if an error was encountered, then the host should send the AVT-716 a disconnect command (\$61 \$01) and attempt the initialization again.

"Fast Initialization" example:

Host sends to 716 unit: 65 13 81 10 F1 81

\$65 - op command, 5 bytes follow

\$13 - fast initialization command

\$81 10 F1 81 is the start communication request message

716 sends to host: 11 xx rr aa bb cc dd ...

\$11 - received message from bus

\$xx - count of bytes to follow

\$rr - received status byte

\$aa bb cc dd ... start communications response from ECU

716 sends to host: 71 03 - op mode report, initialization successful, normal operations.

### Keep Alive Message

Immediately following a successful initialization (using any of the three methods) the "Keep Alive" timer in the AVT-716 unit begins a count down. If approx. 4.5 seconds passes with no transmit message activity, then the 716 will queue up the "Keep Alive" message and begin transmitting it.

If the "Keep Alive" message is deleted then nothing will be sent automatically. If more than 5 seconds passes without any transmit activity the specification requires the communications link be terminated. The AVT-716 will detect this situation, disconnect from the vehicle, switch to the un-initialized state, and send the \$71 \$01 message to the host.

Note that a default "Keep Alive" message is defined. This may be changed or disabled, as desired.

### Message Length

Since the KeyWord Protocol 2000 specification supports messages with as many as 255 bytes of data; the format of communications between the AVT-716 unit and the host 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. Remember that the AVT-716 unit will automatically compute and append the checksum byte.

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-716 are all equivalent. The header byte(s) have been bolded and underlined for clarity.

**0E** A1 B2 C3 D4 E5 F6 A7 B8 C9 DA EB FC AD BE

**11 0E** A1 B2 C3 D4 E5 F6 A7 B8 C9 DA EB FC AD BE

**12 00 0E** A1 B2 C3 D4 E5 F6 A7 B8 C9 DA EB FC AD BE

### Receiving Messages from the Bus

The AVT-716 opens a receive buffer when the start of a message is detected. When the entire message is received the buffer is closed, the checksum computed and compared to the one received, the receive status byte is assembled, and then the whole message is forwarded to the host.

All messages received from the bus may be long as well. All messages received from the bus are passed to the host using the header byte format that is most applicable to the situation. (The length of the received message determines which header format is used.)

Remember that the AVT-716 puts a received status byte on the front of the message. Also remember that the AVT-716 unit computes the checksum and checks it against the one received on the message. The checksum byte is then deleted and is not sent to the host (unless requested using the 52 01 01 command).