VOLKSWAGEN

GROUP OF AMERICA

K-line Communication Description

Introduction

There are two primary ISO documents that describe how to perform OBD communications on K-line between a “tester” and a “vehicle”. There are actually several ISO and SAE documents, but these two are the main ones that describe all aspects of the items to be discussed in this white paper. These two documents also reference the other applicable documents. They are:

- ISO 9141 part 2 and,
- ISO 14230 in four parts, aptly labeled -1 through -4. Parts 2 and 4 are the ones that are pertinent here.

ISO 14230 is also known as Keyword Protocol 2000, or KWP. The term KWP will be used for general references and the specific ISO 14230 part will be used when necessary for clarification.

For this document, the “tester” will be defined as a scan tool or other device connected to the SAE J1962 connector to be used for Inspection and Maintenance to gather OBD data from the vehicle. The “vehicle” will be defined as the car under test and its OBD-related ECU(s).

Initialization

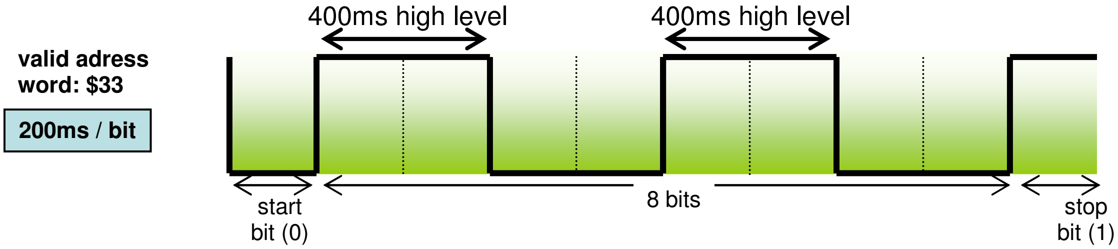
These two documents define three different methods to accomplish asynchronous-to-synchronous communications also known simply as “protocols”. These three protocols are separate and distinct.

1) ISO 9141 with 5-baud initialization (init)
2) KWP with 5-baud init
3) KWP with fast init

In Title 13, California Code Regulations, Section 1968.2(g)(3), paraphrasing, allows manufacturers to use ONE standardized protocol for OBD Communication.

This is where things have been commonly confused. ISO 14230-4, clause 4.4 states that all OBD ECUs on a single vehicle shall only support one of either the 5-baud init OR the fast init. ISO 9141 also defines a 5-baud init sequence, which is differentiated from the KWP 5-baud init sequence by the key bytes that the vehicle responds with. These three protocols overlap in several ways but are still distinct and separate. So, the following is a fairly complete description of each protocol’s initialization scheme that is intended to help tool and equipment manufacturers. All data values are hexadecimal.

1) ISO 9141 5-baud init starts its initialization sequence by the tester issuing the address 33 at 5 bits per second and looks like the following:



The total transmit time for the address lasts for two seconds. After validation of the address internally in the vehicle ECU(s), which takes a time known as W1, which is between 20 and 300 ms long, the vehicle will respond with the synchronization byte 55 informing the tester of the new baud rate which should now be 10.4 kbps. The vehicle shall then wait a time known as W2, which is between 5 and 20 ms, for the tester to reconfigure to the new baud rate, and then the vehicle will send the two key bytes. These key bytes shall be either 08 08, or 94 94, separated by a time known as W3, which is between 0 and 20 ms, that describe to the tester the value of P2MIN to be used. As an acknowledgement of reception of the key bytes, the tester, after waiting a time known as W4, which is between 25 and 50 ms, shall then invert the key byte #2 and send it to the vehicle. After waiting another period equal to W4, the vehicle shall then invert the initialization address of 33 and send it to the tester as the “ready to communicate” signal. This ends the initialization sequence.

2) KWP 5-baud init looks identical to the ISO 9141 5-baud init, except for the key bytes sent from the vehicle to the tester. There are 19 different key byte sets defined for KWP that look like 8F xx, where xx describes the format of the header to be used but only 4 of them (8F E9, 8F 6B, 8F 6D, and 8F EF) are allowed for legislated/required OBD communication by ISO 14230-4, clause 4.4. ISO 14230-4, clause 4.4 further states that only the functionality of 8F E9 is to be used between the tester and the vehicle, regardless of which of the 4 valid key bytes it received from the vehicle. This means that, for all messages, a three-byte header is used, no additional length byte is used, and normal timing is used. This ends the initialization sequence. This is separate and different from the ISO 9141 5-baud init protocol described above only in the valid key bytes that are received from the vehicle (and of course, the inverse of key byte #2 which is then transmitted from tester to vehicle to confirm).

3) KWP fast init is a completely different process from the others. It does not involve any 5-baud communication and works solely on 10.4kbps. The init sequence begins with a “wake-up” pattern transmitted by the tester to the vehicle that is 50 ms long followed immediately by a StartCommunication request from the tester to the vehicle. The vehicle should then respond to the tester with a StartCommunication response.

The StartCommunication request message is comprised of a format byte, target address byte, source address byte, and a service ID byte. Since functional addressing is required for legislated/required OBD communication, the format byte will be C1, the target address will be 33, the source address (tester) will be F1, and the StartCommunication request service ID is 81. Thus the StartCommunication request message is:

C1 33 F1 81 66

with 66 being the checksum. There is no vehicle response allowed between the “wake-up” pattern and the StartCommunication request. The first vehicle response will be a StartCommunication positive response. A StartCommunication response message is similarly comprised of a format byte, a target address byte, a source address byte, a service ID byte, and additionally, the key bytes. The format byte is defined in SAE J1979 Table 13 as - 10 LL LLLLb - where LL LLLL is “Length of data bytes” in the message. Thusly in this example - $83 = 10 00 0011b - meaning that the length of data bytes is 00 0011b or 3 data bytes. This table also defines that there are a maximum of 7 data bytes, therefore, the range of the format byte is between $81 and $87. The target address byte is F1 (tester), the source address is the responding ECU's address (10 in this example), and the StartCommunication response service ID is C1. A positive StartCommunication response message would look something like:

83 F1 10 C1 yy zz cs

Where yy zz are the key bytes as described before and cs is the checksum. A proper vehicle response such as the one above must be received before any subsequent messages on KWP can successfully be transmitted or received.

This ends the initialization sequence. This is separate and different from the KWP or ISO 9141 5-baud init sequences described above.

The separation and differentiation between the three protocols needs to be completely understood when attempting to start OBD communication with a vehicle. The key bytes are the important differentiator during 5-baud initialization attempts to determine whether the ISO 9141 or ISO 14230 protocol is to be used for this communication session.

NOTE: Due to this key byte differentiation, only one 5-baud initialization sequence by the tester is necessary to determine protocol. This can help speed up the initialization process.

There is no prescribed order in which to attempt initialization of communications in either of the ISO documents. SAE J1978 confirms this by stating the tests to determine the protocol may be done in any order and where possible, simultaneously. However, SAE J1978 does specify the following order as one example:

- J1850 PWM
- J1850 VPW
- ISO 9141-2
- ISO 14230-4
- ISO 15765-4

Experience from the field is that, while not required, utilizing this order can minimize many communication initialization problems seen in the field. When properly implemented, however, any order can be done and still achieve successful communication with vehicles.

Error Handling

Some testers try KWP fast init prior to the combined ISO 9141/KWP 5-baud init sequence but make the mistake of sending the 5-baud init sequence too soon after the KWP fast init attempt. This leads to the following failure of initialization:

Once the K-line has been pulled low for the wake-up pattern in KWP fast init, any vehicle ECU on the line will be attempting to either read and validate a 5-baud address or read and validate a KWP fast init with StartCommunication request. If the "pulled low" condition is a KWP fast init and the vehicle ECU(s) actually use ISO 9141 5-baud init, the vehicle ECU(s) may require the entire 2 second address time + W1 time to attempt to validate a 5-baud address before realizing it was not a valid 5-baud init sequence. If the tester follows an unsuccessful KWP fast init attempt with an ISO 9141/KWP 5-baud init before the end of this time [address (2 seconds) + W1 validation period (300 ms) + W5 tester wait period (300 ms)], the vehicle ECU(s) will not be prepared to receive the message and the communication attempt will fail (see below).

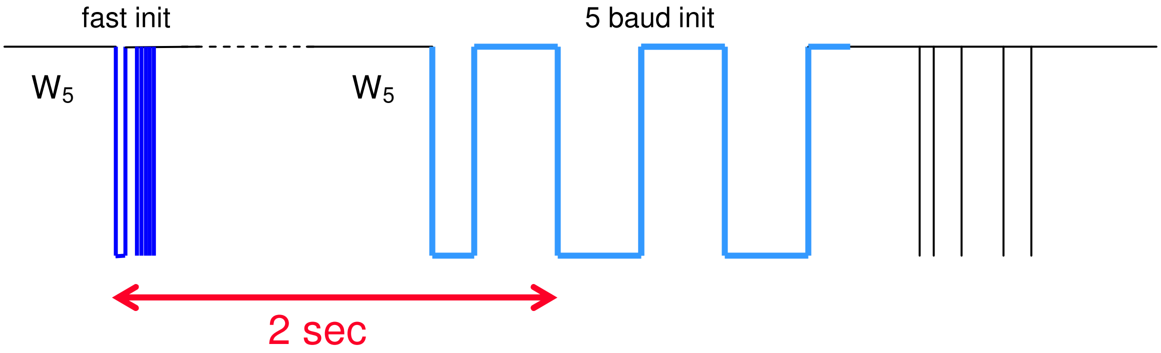
There are two recommended initialization methodologies that will be described here:

1) Attempt a 5-baud initialization, followed by a KWP fast initialization. This guarantees that the correct timings have been maintained.

2) Attempt a KWP fast initialization. If unsuccessful, make sure that a full 2 second address time has passed, along with the 300 ms W1 time and the 300 ms W5 time, for a total time of 2.6 secs from the end of the last transmitted KWP fast init message before beginning the 5-baud init sequence.

A further explanation of the second option is as follows:

Many vehicle ECUs have been designed based on the ISO 9141 protocol prior to the development and introduction of KWP. Several of these ECUs use hardware or software methodologies to validate the 5-baud address that wait for the entire address to be received before such validation takes place. The following diagram shows a KWP fast init followed too closely by a 5-baud init.



If such an ISO 9141 ECU is the recipient of such a sequence, it will recognize an initialization error after the 2 second period shown above, and according to ISO 9141 clause 13.1 “If an ECU detects an error…the ECU will immediately be prepared to recognize the initialization address.” The ECU will detect an incorrect address and will reset and await a new initialization address from the tester at the 2 second point. As shown in the diagram, the tester is now already in the process of performing a K-line 5-baud init. The ECU may see another high-to low transition in the K-line as the next initialization attempt and be reading the next 2 seconds worth of data, which is also erroneous to the ECU and it will reset again. When the tester finishes sending the address, it will get no response from the vehicle and consider this a failed 5-baud init attempt.

Communications will never take place in this scenario. Therefore, even though the tester was attempting a KWP fast init to begin with, it should be aware that an ISO 9141 ECU may have been “awakened” and should wait for the full 2 seconds of a 5-baud address transmission, plus the possible time that such a module might use to finish validating the address (W1), plus the tester wait time described by W5 before attempting a new initialization sequence. Thus, the next initialization attempt on the K-line must not occur until at least 2.6 seconds after the first fast init attempt.

Clause 13.1 of ISO 9141-2 “Error Handling – Initialization” and Clause 6.1 of ISO 14230-2 “Error Handling – StartCommunication Service” both describe exactly how the tester and the vehicle shall perform when an error has been detected. It is also recommended that if subsequent attempts to communicate are made via the K-line, a wait time of P3MAX is used between attempts to ensure that all modules have dropped out of any communication session that may have previously been started.

First Data Exchange

It is also necessary to understand that all of the above describes the complete initialization sequence for each protocol. Any subsequent communication after this will be standard data requests and responses and are not part of initialization. These data requests generally start with a Service 1 PID 00 request for vehicle supported information. The Service 1 PID 00 request that can be made after any 5-baud init sequence must be compliant with the applicable standard and must have the correct header in relation to the responded key bytes from the vehicle. Therefore, only the following should be true:

- For vehicle key bytes 08 08 or 94 94 the only correct and valid Service 1 PID 00 request message is:

68 6A F1 01 00 C4

- For vehicle key bytes (8F E9, 8F 6B, 8F 6D, and 8F EF) the only correct and valid Service 1 PID 00 request message is:

C2 33 F1 01 00 E7

It is well known that some vehicle manufacturers have anomalies related to these items. This may lead to a scan tool manufacturer performing some sequences “above and beyond” the standard prescribed methodologies to make a successful communication attempt. These “above and beyond” attempts should only be made after the standard prescribed methodologies have been exhausted.

For KWP fast init, the only time that a Service 1 PID 00 request can be made is after the vehicle has responded with a StartCommunication positive response. If there is no vehicle StartCommunication positive response, it means that the initialization sequence has failed.

Address Bytes

Physical address values for modules within vehicles have not been assigned by either ISO 9141 or ISO 14230. This is especially true of modules using either of the 5-baud initialization protocols. Annex A.1 of ISO 14230-2 “Physical Addresses” which states that “Addresses are controlled by vehicle manufacturers” along with the ISO 9141 definition of the how the address byte is constructed. For ISO 14230 fast init addresses, Annex B of the same document states “Addresses…may be the same as used for 5 baud initialization or be in accordance with SAE J2178, Part 1”. Note that SAE J2178 is a standard for J1850 Class B network communications. Accordingly, as with J1850, testers should not ‘expect’ particular vehicle ECU addresses or make assumptions about the role of a vehicle ECU based on its address. For example, engine ECUs commonly use 10 as an address but there are many vehicles that do not use 10 for the engine ECU or may use 10 for another vehicle ECU.

Data Collisions

Data collisions can occur during normal communication on the K-line. These cases are rare and are easily handled. When a data collision occurs, the ECU’s in the vehicle will employ an arbitration methodology similar to that described in SAE Technical Paper 950041. In all cases where the tester detects an error in the vehicle response for K-line protocols, the tester shall retransmit the original request. The ECU’s at this point should have performed the given arbitration method and will now return correct responses. From this point, communication can continue normally.