INTERNATIONAL STANDARD

ISO 14230-2

First edition
1999-03-15

Road vehicles — Diagnostic systems — Keyword Protocol 2000 —

Part 2:
Data link layer

Véhicules routiers — Systèmes de diagnostic — Protocole «Keyword 2000» —

Partie 2: Couche de liaison de données

Reference numbe
ISO 14230-2:199 (E)

Contents

Page

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet iso@iso.ch

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard ISO 14230-2 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, subcommittee SC 3, *Electrical and electronic equipment*.

ISO 14230 consists of the following parts, under the general title *Road vehicles — Diagnostic systems — Keyword Protocol 2000* :

— *Part 1: Physical layer*

— *Part 2: Data link layer*

— *Part 3: Application layer*

— *Part 4: Requirements for emissions-related systems*

Annex A forms an integral part of this part of ISO 14230. Annexes B and C are for information only.

Introduction

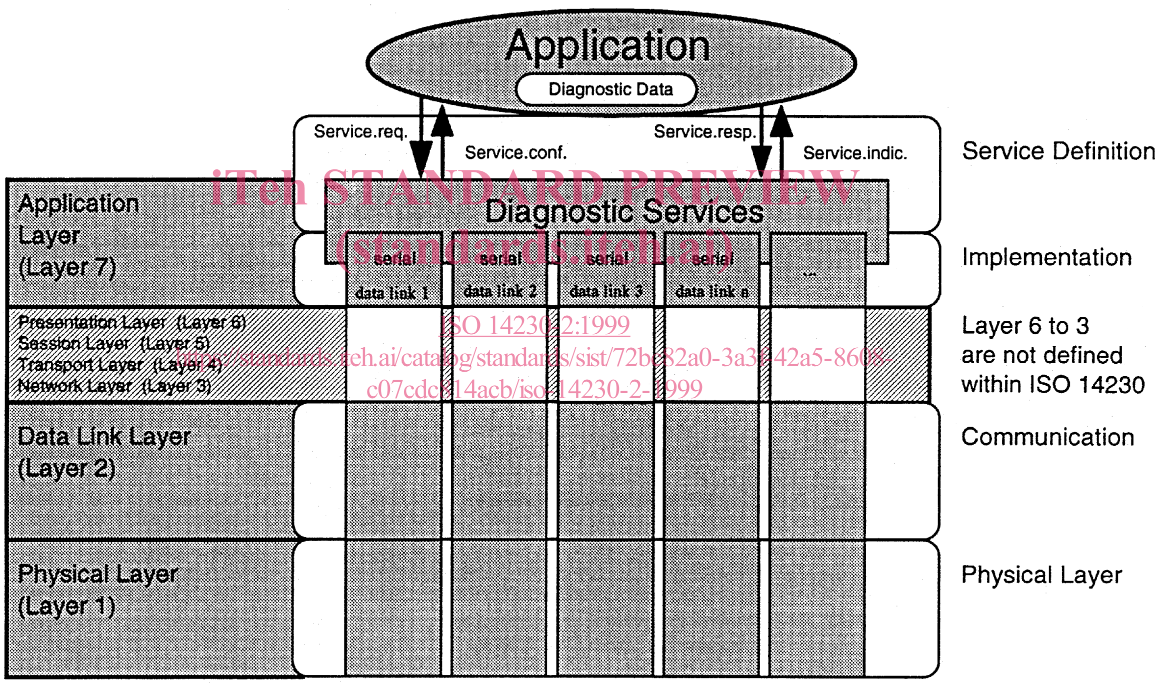
ISO 14230 has been established in order to define common requirements for diagnostic systems implemented on a serial data link.

To achieve this, it is based on the Open System Interconnection (OSI) Basic Reference Model in accordance with ISO 7498 which structures communication systems into seven layers. When mapped on this model, the services used by a diagnostic tester and an Electronic Control Unit (ECU) are broken into:

— diagnostic services (layer 7);

— communication services (layers 1 to 6),

in accordance with figure 1.



Example of serial data links: KWP2000, VAN, CAN, J1850, etc.

Figure 1 — Mapping of diagnostic services on OSI Model

Road vehicles — Diagnostic systems — Keyword Protocol 2000 —

Part 2:
Data link layer

1 Scope

This part of ISO 14230 specifies common requirements of diagnostic services which allow a tester to control diagnostic functions in an on-vehicle Electronic Control Unit (for example, electronic fuel injection, automatic gearbox, antilock braking system, etc.) connected on a serial data link embedded in a road vehicle.

It specifies only layer 2 (data link layer). Included are all definitions which are necessary to implement the services (described in ISO 14230-3) on a serial link (described in ISO 14230-1). Also included are some communication services which are needed for communication/session management and a description of error handling.

This part of ISO 14230 does not specify the requirements for the implementation of diagnostic services.

The physical layer may be used as a multiuser-bus, so a kind of arbitration or bus management is necessary. There are several proposals which are not part of this part of ISO 14230. The car manufacturers are responsible for the correct working of bus management.

Communication between ECUs are not part of this part of ISO 14230.

The vehicle diagnostic architecture of this part of ISO 14230 applies (see figure 2) to

— a single tester that may be temporarily or permanently connected to the on-vehicle diagnostic data link, and

— several on-vehicle electronic control units connected directly or indirectly.

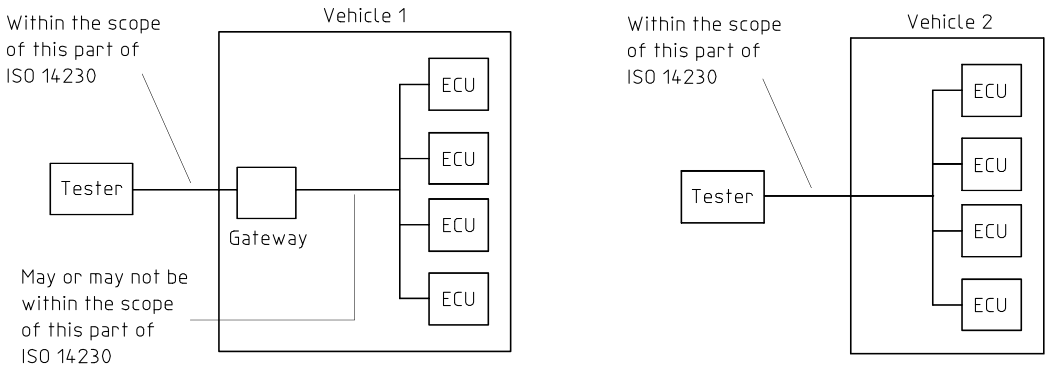


Figure 2 — Vehicle diagnostic architecture

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 14230. All the time of publication, the editions were indicated were valid. All standards are subject to revision, and parties to agreement based on this part of ISO 14230 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 9141:1989, *Road vehicles — Diagnostic systems — Requirements for interchange of digital information.*

ISO 9141-2:1994, *Road vehicles — Diagnostic systems — Part 2: CARB requirements for interchange of digital information.*

ISO 14230-3:1999, *Road vehicles — Diagnostic systems — Keyword Protocol 2000 — Part 3: Application layer.*

ISO 14230-4:1999, *Road vehicles — Diagnostic systems — Keyword Protocol 2000 — Part 4: Requirements for emission related systems.*

SAE J 1979: 1996, *E/E diagnostic test modes.*

3 Physical topology

Keyword Protocol 2000 is a bus concept. Figure 3 shows the general form of this serial link.

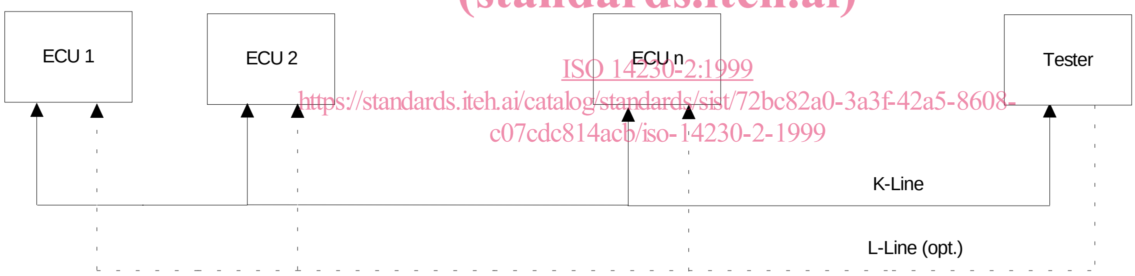


Figure 3 — Topology

The K-line is used for communication and initialization; the L-line (optional) is used for initialization only. Special cases are node-to-node-connection, which means there is only one ECU on the line which also can be a bus converter.

4 Message structure

4.1 General

The message structure consists of three parts:

— header;

— data bytes;

— checksum.

See figure 4.

Header and checksum bytes are described in this part of ISO 14230. The area of data bytes always begins with a Service Identification.

The data bytes and their use are described in ISO 14230-3 and this part of ISO 14230.

Header				Data bytes		Checksum
Fmt	Tgt[1)]	Src[1)]	Len[1)]	SId[2)]	. . Data[2)] . .	CS
max . 4 byte				max. 255 byte		1 byte

1) bytes are optional, depending on format byte.

2) Service Identification, part of data bytes.

NOTE — The shaded area (header, checksum) are described in this part of ISO 14230.

Figure 4 — Message structure

4.2 Header

The header consists of a maximum of 4 bytes. A format byte includes information about the form of the messages, target and source address bytes are optional for use with multinode connections, an optional separate length byte allows message lengths up to 255 bytes.

4.2.1 Format byte

The format byte contains 6 bit length information and 2 bit address mode information. The tester is informed about use of header bytes by key bytes (see 4.2.2).

A1	A0	L5	L4	L3	L2	L1	L0

where

A1 and A0 define the form of header which will be used by the message in accordance with table 1;

L5..L0 define the length of a message from the beginning of the data fields (ServiceIdentification byte included) to checksum byte (not included). A message length of 1 to 63 bytes is possible. If L0 to L5 = 0 then the additional length byte is included (see 4.2.5).

Table 1 — Header message form

A1	A0	Mode
0	0	no address information
0	1	Exception mode (CARB)
1	0	with address information, physical addressing
1	1	with address information, functional addressing

A.1,A.0=01 (CARB mode) is an exception mode. The CARB mode is not specified in this part of ISO 14230. CARB uses format bytes $68 (0110 1000) and $48 (0100 1000). For more details refer to ISO 9141-2 and SAE J1979.

4.2.2 Target address byte

This is the target address for the message and is always used together with the source adress byte. The target address in the request messages sento to the ECU may be a physical or a functional address. The target address in the response messages sent to the tester shall be the physical address of the tester. Physical addresses may be the 5 baud address byte (see annex A) or addresses according to SAE J 2178-1 (see annex B). The target address byte is optional and only necessary on multimode bus topologies. For node-to-node connections it may be omitted. For CARB messages this byte is defined in ISO 9141-2 or ISO 14230-4.

4.2.3 Source address byte

This is the address of the transmitting device. It shall be a physical address. There are the same possibilities for the values as described for physical target address bytes. Addresses for testers are listed in SAE J2178-1 (see annex B). This byte is optional (always used together with target address byte) and only necessary on multinode bus topologies. For node-to-node connections it may be omitted.

4.2.4 Length byte

This byte is provided if the length in the header byte (L0 to L5) is set to 0 as shown in table 2. It allows the user to transmit messages with data fields longer then 63 bytes. With shorter messages it may be omitted. This byte defines the length of a message from the beginning of the data field (service identification byte included) to checksum byte (not included). A data length of 1 byte to 255 bytes is possible. The longest message consists of a maximum of 260 bytes. For messages with data fields of less than 64 bytes there are two possibilities: length may be included in the format byte or in the additional length byte. An ECU does not need to support both possibilities, the tester is informed about the capability of an ECU through the keybytes (see 6.1.2.1).

Table 2 — Presence of a length byte

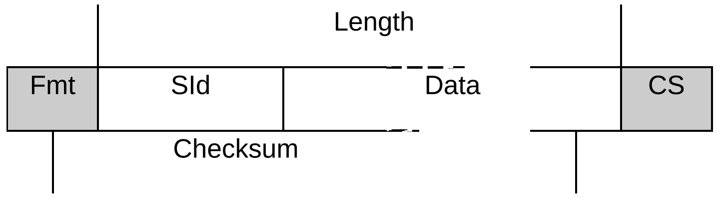
Length	Length provided in	
	Fmt byte[1)]	Length byte
< 64	XX00 0000	present
< 64	XXLL LLLL	not present
≥ 64	XX00 0000	present

1) XX : 2 bits address mode information (see 4.2.1)

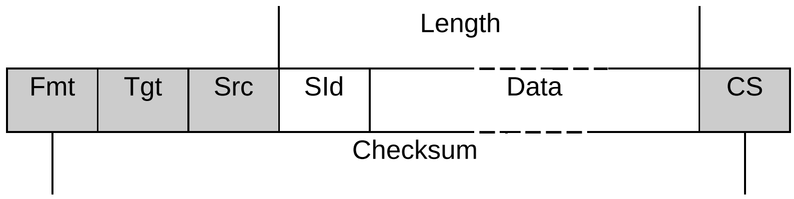
LL LLLL : 6 bits length information.

4.2.5 Use of header bytes

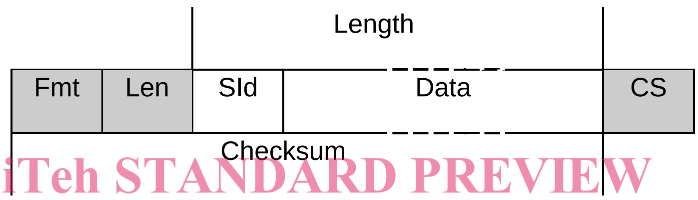
With the above definitions there are four different forms of message. These are shown diagramatically in figure 5.



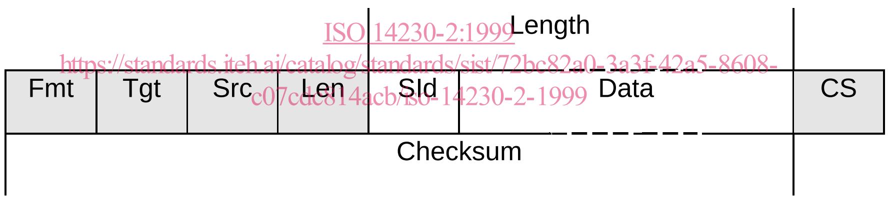
a) Header with address information, no additional length byte



b) Header with address information, no additional length byte



c) Header without address information, additional length byte



d) Header with address information, with additional length byte

Fmt	Format byte	SId	Service Identification Byte
Tgt	Target address (optional)	Data	(depending on service)
Src	Source address (optional)	CS	Checksum byte
Len	additional length byte (optional)		

NOTE — The unshaded area is defined in ISO 14230-3.

Figure 5 — Header messages

4.3 Data bytes

The data field may contain up to 63 or up 255 bytes of information, depending on the use of length information. The first byte of the data field is the Service Identification Byte. It may be followed by parameters and data depending on the selected service. These bytes are defined in ISO 14230-3 (for diagnostic services) and in clause 5 (for communication services).

4.4 Checksum byte

The checksum byte (CS) inserted at the end of the message block is defined as the simple 8 bit sum series of all bytes in the message, excluding the checksum.

If the message is

$\langle 1\rangle \langle 2\rangle \langle 3\rangle .. \langle N\rangle, \langle CS\rangle$

the two following cases may occur:

when $\langle i\rangle$ $(1 \le i \le N)$ is the numerical value of the i^{th} message byte, then :

$\langle CS\rangle = \langle CS\rangle_N$

when $\langle CS\rangle_i$ $(2 \le i \le N)$:

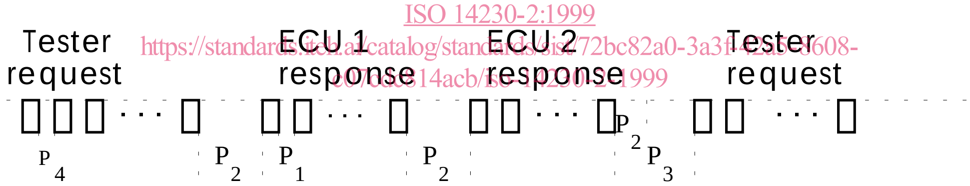
when $\langle CS\rangle i = \{ \langle CS\rangle_{i-1} + \langle i\rangle \}$ mod256 et
$\langle CS\rangle_1 = \langle 1\rangle$

Additional security may be included in the data field as defined by the manufacturer.

4.5 Timing

4.5.1 Value entering

During normal operation the timing parameters as shown in figure 6 are relevant.



Value	Description
P1	Inter byte time for ECU response
P2	Time between tester request and ECU response or two ECU responses
P3	Time between end of ECU responses and start of new tester request
P4	Inter byte time for tester request

Figure 6 — Timing

There are two sets of default timing parameters :

a) one set for normal functional and physical addressed communication. Longer timings are necessary to allow any technics of bus management;

b) one set restricted to physical addressing to allow faster communication.

The tester is informed about the capability of an ECU through the keybytes (see 5.2.4.1).

Timing parameters may be changed with the communication service "AccessTimingParameters" (see 5.4).

Users shall take note of limits listed below and the following restrictions:

$P3_{min} > P4_{min}$

$Pi_{min} < Pi_{max}$ for i = 1, .., 4

There may be further restrictions, when the tester and listening ECUs detect the end of a message by timeout. In this case the following restrictions are valid:

$P2_{min} > P4_{max}$

$P2_{min} > P1_{max}$

In case of functional addressing, i.e. that there may be more than one response to one request, further restrictions may be added.

It is in the designers' responsibility to ensure proper communication in the case of changing the timing parameters from the default values.They shall also ensure that the chosen communication parameters are possible for all ECUs which participate in the session.

The possible values depend on the capabilities of the ECU. In some cases the ECU may need to leave its normal operation mode to switch over to a session with different communication parameters.

Tables 3 and 4 show the timing parameters which are used as default, the limits within which they can be changed and the resolution which may be used to set a new value (with the communication service AccessTimingParameter: see 5.4).

Table 3 — Normal Timing Parameters Set (for functional and physical addressing)

Values in milliseconds

Timing Parameter	Minimum values			Maximum values		
	Lower limit	Default	Resolution	Default	Upper limit	Resolution
P1	0	0	-	20	20	-
P2	0	25	0,5	50	See table 5	
P3	0	55	0,5	5 000	∞ ($FF)	250
P4	0	5	0,5	20	20	-

Table 4 — Extended Timing Parameters Set (for physical addressing only)

Values in milliseconds

Timing Parameter	Minimum values			Maximum values		
	Lower limit	Default	Resolution	Default	Upper limit	Resolution
P1	0	0	-	20	20	-
P2	0	25	0,5	50	See table 5	
P3	0	55	0,5	5 000	∞ ($FF)	250
P4	0	5	0,5	20	20	-

Table 5 — P2max Timing Parameter calculation

Hex. value	Resolution	Maximum value ms	Maximum value calculation method ms
01 to F0	25	25 to 6 000	(hex. value) x (Resolution)
F1	see maximum value calculation method	6 400	(low nibble of hex. value) x 256 x 25 Example of $FA : ($0A x $0100) x 25 = 64 000
F2		12 800	
F3		19 200	
F4		25 600	
F5		32 000	
F6		38 400	
F7		44 800	
F8		51 200	
F9		57 600	
FA		64 000	
FB		70 400	
FC		76 800	
FD		83 200	
FE		83 600	
FF	-	∞	Not applicable
The P2max timing parameter value shall always be a single byte value in the AccessTimingParameter service. The timing modifications shall be activated by implementation of the AccessTimingParameter service.			

Proposed $P2_{max}$ timing parameter calculation method (values > 6 000 ms):

The $P2_{max}$ timing parameter calculation uses 25 ms resolution in the range of $01 to $F0. Beginning with $F1, a different calculation method shall be used by the server and the client in order to reach $P2_{max}$ timing values greater than 6 000 ms.

Calculation Formula $P2_{max}$ values > $F0

$$\text{Calculation\_Of\_}P2_{max} = (\text{low nibble of } P2_{max}) \times 256 \times 25 \text{ (ms)}$$

The P2max timing parameter value shall always be a single byte value in the AccessTimingParameter service. The timing modifications shall be activated by implementation of the AccessTimingParameter service.

4.5.2 Timing exceptions

The extended P2 timing window is a possibility for (a) server(s) to extend the time to respond on a request message. A P2max timing exception is only allowed with the use of one or multiple negative response message(s) with response code $78 (RequestCorrectlyReceived-ResponsePending) by the server(s). This response code shall only be used by a server in case it cannot send a positive or negative response message based on the client's request message within the active P2 timing window. This response code shall manipulate the P2max timing parameter value in the verver and the client. The P2max timing parameter is set to the value (in ms) of the P3max timing parameter. The client shall remain in the receive mode. The server(s) shall send multiple negative response messages with the negative response code $78 if required.

As soon as the server has completed the task (ourtine) initiated by the request message it shall send either a positive or negative response message (negative response message with a response code other than $78) based on the last request message received. When the client has received the response message which has been preceeded by the negative response message(s) with response code $78, the client and the server shall reset the P2max timing parameter to the previous timing value. The client shall not repeat the request message after the reception of a negative response message with response $78 .