WABCO	SPECIFICATION E677A.DOC	99-03
	CLASS: TEST INSTRUCTION, CONDITIONS OF DELIVERY	CLASS No.: 61
	DIAGNOSIS BY SERIAL DATA EXCHANGE	JED 677
PREV. EDITION 95-03	ABBREVIATED TITLE Diagnosis, ser. data exch.	852 006 770 4

1 Scope

The specification refers to provisions for diagnosis by serial data exchange.

JED-677 specifies a uniform standard for all WABCO electronic systems.

2 General

The diagnosis by serial data exchange as specified in this document exclusively refers to devices which are able – due to their "inherent intelligence" – to test the system periphery and/or detect functional defects and to control a disturbance lamp if necessary.

In a first step, the workshop staff shall be informed of the reasons for the disturbance by means of a service device. In a second step, the operator shall be instructed how to do the repair using simple measuring means. For the production at the original equipment manufacturer, service devices are required which allow functional tests and recording of manufacturing data.

In conformance with the Product Specification, WABCO vehicle electronic systems perform the following tasks through serial data exchange with a service device:

- transfer of vehicle-specific data into an EEPROM
- examination and correction of stored data
- error diagnosis with repair instruction
- inspections in workshop
- output of service period information
- output of current parameters and sensor data
- control of the storage of instantaneous vehicle data in order to compensate for tolerances (calibration)
- control of functional sequences for test purposes
- determination of wear

In addition to the interface requirements and the provisions for bidirectional data communication to realize the tasks mentioned in chapter 2 with an external service device, the requirements for an on-board diagnosis according to the free-running principle have been included in this specification.

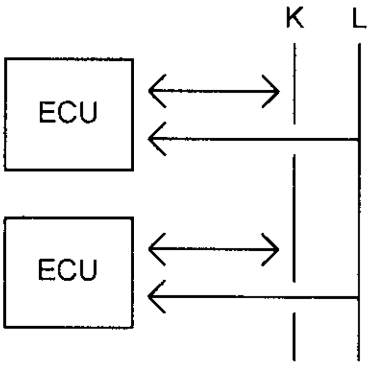
continuation page 2 to 25

3 Specification of the serial interface

3.1 Serial interface to ISO 9141

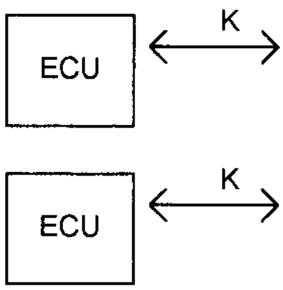
The system configuration type 8 to ISO 9141 shall be used as the diagnostic interface for bus systems, type 8 or type 5 for single-line systems.

Bus system
type 8 to ISO 9141



- bidirectional line K
- unidirectional line L

Single-line system
type 5 to ISO 9141



- bidirectional line K

Line K : bidirectional data communication between tester and ECU.

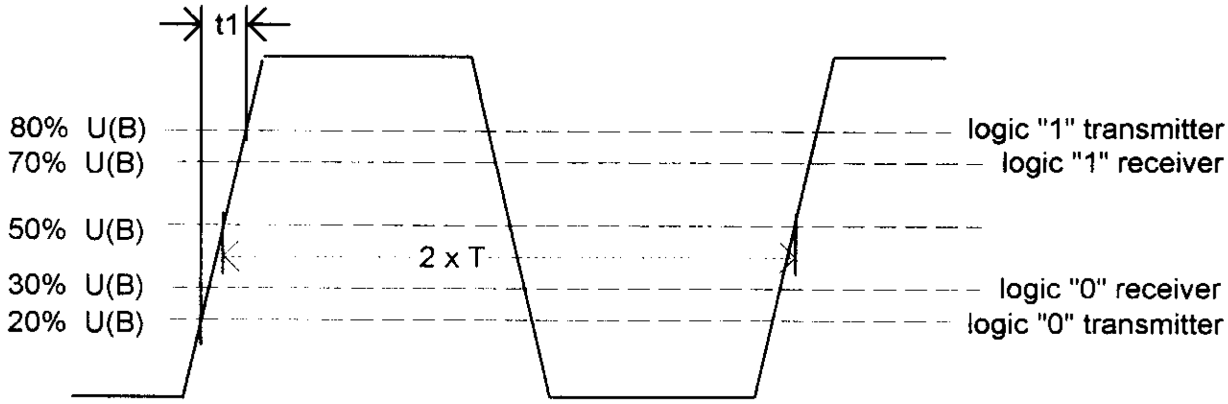
Line L : unidirectional data transfer from the tester to the ECU (command transmission).

After completion of the initialization the application of line L depends on the ECU. Line L is provided for auxiliary functions, such as the control of the transfer direction on line K.

3.2 Signal specification

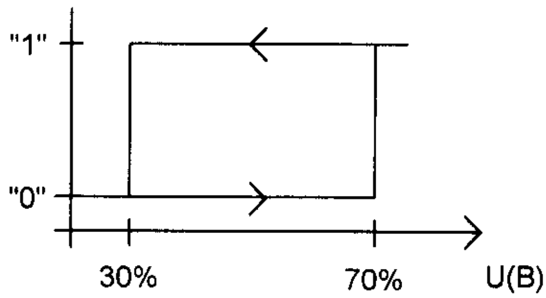
Transmitter:	signal logic "0"	$\rightarrow$	U(Low) $\leq$ 20 % U(B)
	signal logic "1"	$\rightarrow$	U(High) $\geq$ 80 % U(B)
Receiver:	signal logic "0"	$\rightarrow$	U(Low) $\leq$ 30 % U(B)
	signal logic "1"	$\rightarrow$	U(High) $\geq$ 70 % U(B)

12 V systems:	8 V $\leq$ U(B) $\leq$ 16 V
24 V systems:	16 V $\leq$ U(B) $\leq$ 32 V



t1 < 10 % T T = time period for one bit

The slope times t1 shall be less than 10 % of the bit time. The bit time is defined as half of the time between the 50 % levels of successive rising and falling edges of alternating "1" and "0" bits. The slope times t1 are defined as the time taken for the voltage to change from 20 % to 80 %, and 80 % to 20 % U(B).

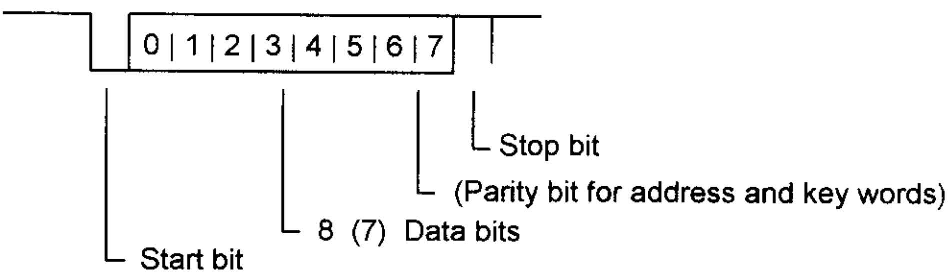


3.3 Transmission speed and data format

Data transmission on line K within the range from 200 to 10000 baud (deviation from the adjusted baud rate max. ± 1 %). Transmission of the address code for initialization of the ECU on lines L and K with 5 baud (deviation max. ± 0.5 %).

The system-specific baud rate has to be particularly emphasized in the Product Specification.

Data are transmitted according to the NRZ procedure with start bit and stop bit as well as a parity bit (odd parity) for the 5 baud address and the key words. The transmission of a byte starts with the LSB.



3.4 Electrical specifications

The values are indicated for a system consisting of max. n = 10 electronic units. The values in parentheses refer to 24 V systems.

Inputs and outputs of the control units (ECU) :

Line L :
- unidirectional
- input resistance to ground
  R(In) $\geq$ 50 kΩ corresponds to 5 kΩ × n
  ($\geq$ 100 kΩ corresponds to 10 kΩ × n)

Line K :
- bidirectional
- if the ECU is inactive, the output transistor is inhibited.
- input resistance to ground
  - at logic "1"
    R(In) $\geq$ 50 kΩ corresponds to 5 kΩ × n
    ($\geq$ 100 kΩ corresponds to 10 kΩ × n)
  - at logic "0"
    R(In) $\leq$ 110 Ω
    ($\leq$ 220 Ω)

Lines K and L:
- pull-up resistance; if an internal pull-up resistance is applied, it shall not be less than 10 kΩ (20 kΩ); this value does not conform to ISO 9141; in practice, this simply means an increase in the drive performance by 4 %
- electromagnetic compatibility in accordance with the Product Specification
- short-circuit proof to
  - ground
  - supply voltage

Inputs and outputs of the tester:

Lines K, L:
- pull-up resistance 510 Ω (1 kΩ)
- leakage resistance > 100 kΩ
- no voltages > U(B) or < ground
- open-circuit-voltages
  - logic "1" > 95 % U(B)
  - logic "0" < 10 % U(B)

Admissible capacitances (lines K and L):

The admissible total capacitance of the system depends on the transmission speed :

$$BR \leq \frac{10^{-4}}{\sum_{i=1}^{n} C(ECUi) + C(OBW) + C(TE)}$$

BR	baud rate; for 24 V systems half of the calculated baud rate shall be applied
C(ECUi)	contribution of an ECU
C(OBW)	input capacitance of the bus lines to ground
C(TE)	contribution of the diagnostic tester and its cables

The value for C(TE) shall not exceed 2 nF.

The admissible capacitance of an ECU is calculated as follows:

12 V system :

$$C(ECU) = \frac{(\frac{10^{-4}}{BR}) - C(OBW) - 2 \times 10^{-9}}{n}$$

24 V system:

$$C(ECU) = \frac{(\frac{0,5 \times 10^{-4}}{BR}) - C(OBW) - 2 \times 10^{-9}}{n}$$

With a 12 V (24 V) system with 10 ECUs and a transmission speed of 9600 baud, the admissible individual capacitance is approx. 640 pF (120 pF) per ECU based on the assumption that the total capacitance of the bus line C(OBW) is 2 nF.

4 Unidirectional free-running diagnosis

Free-running diagnosis can be provided for ECUs that are not connected to a bus. In this unidirectional "monitor mode" the ECU continuously outputs diagnostic data and measuring values via line K during normal operation.

4.1 Protocol

In the free-running mode each electronic unit continuously outputs data according to the protocol shown in Appendix B. The data frames are transmitted sequentially with a maximum cycle time of 0,5 s. Since an erroneous transmission does not cause a malfunction, and the data are transmitted successively several times, the data are not checked for transmission errors.

4.2 Error codes

In addition to the transmission of measuring values and of the contents of the error memory, the ECU explicitly signals a present malfunction by transmission of an error code. The error code is determined according to the importance of the error:

Error unimportant to the safety	1-99
Drive to next workshop still possible	100-199
Serious errors, drive to be interrupted immediately	200-250
Codes	251-255

Code	Meaning
251	Reserved
252	Acknowledgement of switch over to the bidirectional service mode
253	Reserved
254	Reserved
255	Reserved

Remark:
The meanings of the error codes are provided in the Product Specification relevant to the ECU.

4.3 Clearance of the error memory

A special command is provided in the bidirectional service mode for the clearance of the error memory (EEPROM) of an ECU (see section 5).

4.4 Switch over to the service mode

An ECU is switched over to the service mode (initialization) as specified in ISO 9141 by setting the line K to logic "0" for a duration of 1,8 s ± 0,01 s or by transmitting the corresponding 5 baud address. The respective ECU acknowledges the switch over by sending a data frame with the code 252d instead of an error code according to the free-running protocol without the error memory contents and measuring values.

The service mode (handshake procedure) starts with the completion of the transmission of the last byte of this frame. If an ECU does not send data in the free-running mode, the tester starts with the service mode immediately after having received the second key word and after a period of $t \geq 2$ ms.

An ECU must not be initialized for the service mode before the current data frame has been completely transmitted.

The service mode is brought to an end when another 5 baud address is transmitted, when the corresponding command is transmitted by the test unit or when there has been no data exchange for a certain period of time (see section 5).

The unidirectional free-running mode starts automatically after a period of $t \geq 20$ ms (frame synchronization time; see Appendix B) or after turning off and on the ignition. Corresponding provisions shall be given in the Product Specification.

4.5 Electrical specification

The serial interface corresponds to system configurations type 8 or type 5 as specified in ISO 9141 (see section 3).

5 Bidirectional service mode

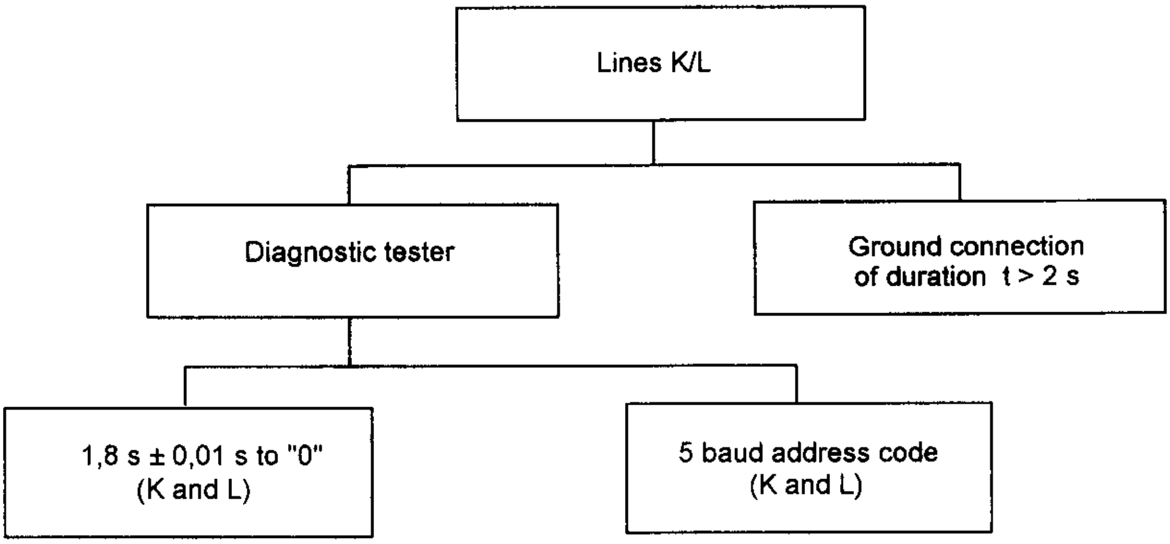
In the following, essential fundamental functions are defined in addition to the hardware requirements. Among the provided command codes only those shall be selected which are relevant to the ECU.

In order to provide appropriate functions for all electronic units, the restrictions resulting from the protocol and from the structure of the command codes are as few as possible.

5.1 Initialization and communication

An ECU transmitting data in the unidirectional free-running mode must not be initialized before the complete transmission of the current data frame (see section 4).

In accordance with ISO 9141 the following procedures to initialize an ECU via line K and L have been determined:



In case of bus systems, the ECU shall be initialized by transmission of a 5 baud address code on lines K and L by a tester (see Appendices C and E). The ECU is able recognize the address code on both lines or only on one of the two lines.

If an ECU is initialized it transmits a synchro byte to synchronize the baud rate, and at least two company-specific key words (see Appendix D). Then, using the command "IDENT", the tester requests the transmission of further bytes with WABCO data (date of production, software version number, product number and status byte). The tester uses these data to select the corresponding service program.

5.2 Leaving the service mode

The ECU must leave the service mode, if

- the tester is switched off or disconnected or there has been no data transmission between the tester and the ECU for a certain period of time;
- the corresponding command has been transmitted by the tester,
- another 5 baud address has been received.

If there is a further initialization while the ECU is in the service mode, the communication is set up as described in section 5.1.

If there is no data communication, NOP commands are transmitted to the ECU by the tester after having reached time-out 1 in order to maintain the communication. If there is no response from the ECU, the tester issues a message and tries to initialize the ECU again. After a NOP command the tester does not transmit further commands or data to the ECU, until time-out 1 is reached. During this period, the electronic unit can start to transmit data to the tester in the handshake procedure.

If the ECU does not receive any further command or data bytes from the tester, it must leave the service mode automatically after reaching time-out 2 even if it is transmitting data to the tester without handshake (see section 5.3.3).

5.3 Data transmission and data safety

In order to guarantee correct data transmission and/or the recognition of disturbances on the transmission line, a handshake procedure is used. An exception are the commands after which the tester does not transmit any data to the ECU for a certain period of time; thus the ECU is able to transmit data to the tester. These data are not transmitted in the handshake procedure.

It is however required that there is a time period t of $T0 \leq t < T1+T0$ between two messages (see section 5.3.2). Thus, after the request for data by the tester, an ECU must wait for the end of time-out 0 to start the transmission of data before time-out 1 has been reached. The first data byte must be transmitted by the ECU before time-out 1 has been reached. Otherwise, it must wait for the next NOP command from the tester or transmit BUSY commands to the tester in order to maintain the communication and the transmission state.

The procedure using BUSY commands should be preferred.

5.3.1 Time-out periods

The time-out periods are used to communicate transmission errors using the handshake procedure and to mutually control the receiving and transmitting readiness. The time-out periods are coupled to the transmission rate so that they can also be maintained with low baud rates without being too long for high transmission speeds.

The following time-out periods are provided:

Time-out 0: $t \geq T0 = 100$ / baud rate

The time-out period is monitored by the tester and the ECU.
The transmitter starts the time-out period if
- one byte was transmitted
- a transmission error was recognized by comparison with the byte returned by the receiver
- time-out 0 has expired once after the recognition of a transmission error.

The transmitter stops the time-out period if the previously transmitted byte was returned by the receiver.

The receiver starts the time-out period after
- the return of the previously received byte
- the receipt of the EXECUTE command.

The receiver stops time-out 0 after having completely received the next byte.

Time-out 1: $t \geq T1 = 300$ / baud rate

The time-out period 1 is monitored only by the tester.
The tester starts the time-out period, if
- the key words were received by the tester after initialization of the ECU.
- time-out 0 has expired and is not started again.

The tester stops time-out 1 if it
- transmits the next byte to the ECU
- has received a byte from the ECU in the handshake procedure

Time-out 2: $t \geq T2 = 400$ / baud rate

The time-out period 2 is monitored only by the ECU.

The ECU starts the time-out period, if

- the key words were transmitted after initialization of by tester
- time-out 0 has expired and is not started again.

The ECU stops time-out 2, if it

- has received the next byte from the tester
- starts transmitting data to the tester in the handshake procedure.

> Note:
>
> The specified time-out periods should be considered as basic setting. Should the tester or the ECU require other time-out periods than those specified, they have to be determined by introducing a factor that is the same for all time-out periods.
>
> Time-out x: $Tx = (a(x) / \text{baud rate}) \times \text{factor}$
>
> $x = \{0, 1, 2\}$ $a = \{100, 300, 400\}$
>
> The time-out periods used shall be particularly emphasized in the Product Specification. If necessary, the tester has to be adjusted to the time-out periods before starting the diagnosis.

5.3.2 The Handshake Procedure

The command bytes and data bytes sent by the transmitter must be returned by the receiver before time-out 0 has been reached. The next data byte is not transmitted until the preceding byte has been received correctly. In this case, time-out 0 must not be exceeded either.

The receiver interprets a time-out 0 as a recognized transmission error and waits for the faulty byte to be repeated after time-out 0 has expired again. Thus the receiver has the possibility to recognize time-out 0 within $T0 \leq t < 2 \times T0$ and to be prepared for the repetition of the faulty byte.

If the ECU, as the receiver, does not repeat the byte sent by the tester, the tester tries to initialize the ECU again. If a byte transmitted by the ECU in the handshake procedure is not repeated by the tester, the ECU switches to reception and waits for commands from the tester, which are not transmitted before time-out 0 has been reached again.

An exception is the EXECUTE command which completes each data transmission according to the handshake procedure. The receiver does not return this command to the transmitter. The transmitter interprets a time-out 0 after transmission of the EXECUTE command as correct completion of data transmission. In case of a transmission error, however, the byte is returned by the receiver before time-out 0 has expired. Now the transmitter repeats the transmission as described above. Therefore, after an EXECUTE command, neither the tester nor the ECU is alllowed to start with the next data transmission before expiration of time-out 0.

The tester, being the transmitter, repeats a faulty byte for a maximum of 10 times before it reports an error and stops the data transmission. As the receiver, the tester stops the transmission of data bytes from the ECU if it has recognized a series of 10 time-out 0. It does not return the next data byte from the ECU and sends a corresponding error message. In both cases the ECU switches to reception and finishes the diagnosis after time-out 2 has expired.

Appendix F shows examples of the handshake procedure.

5.3.3 Data transmission without handshake

After a command which allows the ECU to send data to the tester without handshake, the tester does not send any further data or commands to the ECU before time-out 1 has been reached. Thus, the ECU is able to transmit data without handshake to the tester for the period t = T1 after time-out 0 has expired. However, data transmission has to be completed before this period has expired. The first byte must include the number n of the following data bytes as follows: x = 80 h + n.

If time-out 1 has been reached and the ECU has not transmitted data or stopped the transmission of data, the tester sends a NOP command or the next command to be executed to the ECU using handshake.

If the ECU needs a longer period of time before data can be sent to the tester or if there are a number of data frames which have to be transmitted, the ECU must send BUSY commands to the tester before the transmission and/or between the transmission of individual data frames in order to maintain the communication and the transmission state. These BUSY commands are returned by the tester using handshake, thus time-out 1 and time-out 2 are reset. The tester is able to stop the waiting for data and/or transmitting data by generating a time-out 0.

If time-out 1 is exceeded by the ECU during the transmission of data, the tester will initialize the ECU again.

Examples for data transmission without handshake are shown in Appendix G.

5.4 Set of commands

5.4.1 Basic structure of the command code

The set of commands consists of 1-byte and n-byte commands as well as one special command. The maximum length of a command can be n = 129 bytes.

1st Byte	Meaning
0xxxxxxx	1-byte command code 00h - 7Fh
1yyyyyyy	n-byte command code; the number of bytes (incl. command code, address and data, excl. 1st byte and acknowledgement EXECUTE) is determined by yyyyyyy
10000000	EXECUTE; special command to acknowledge a data transmission by the transmitter

5.4.2 Command code assignment

00h - 0Fh	Administration commands
10h - 1Fh	Standard functions
20h - 3Fh	System function 0 - 31 will be executed
40h - 5Fh	System data 0 - 31 will be read
60h - 6Fh	System-specific program 0 - 15 will be started
70h - 7Fh	System-specific program 16 - 31 will be started; the tester waits for data sent by the ECU without handshake (see section 5.3.3)

The meanings of the 1-byte command codes and the n-byte command codes (2nd byte of the message) are identically assigned. Therefore it is possible to add further parameters, if this is necessary. The only exception is the BUSY command (see 5.4.4). The number and meaning of the parameters shall be provided in the Product Specification.

5.4.3 Special command to acknowledge data transmission

Abbreviation: EXECUTE
Comm. code: [80h]
Description: Data transmission acknowledgement by the tester incase of a transmission tester ➔ ECU or by the ECU in case of a transmission ECU ➔ tester.
Remark: The receiver does not return the command code to transmitter.

5.4.4 Description of the administration commands

Abbreviation: NOP
Comm. code: [00h]
Description: Maintaining communication whenever there is no data exchange.
Example: [00h] [EXECUTE]
Remark: The NOP command may only be used by the tester.
After an NOP command form the tester the ECU must begin with a data transmission using handshake before time-out 1 has been reached. The first data byte must have been sent before time-out 1 has been reached.

Abbreviation: BUSY
Comm. code: [01h]
Description: Maintaining communication and transmission state whenever there is no data exchange.
Example: [01h] [EXECUTE]
Remark: After a BUSY command from an ECU to the tester the ECU must start the next message before time-out 1 has been reached. The tester uses the BUSY command for maintaining communication when data transmission from the ECU is not permitted (cf. NOP command). Parameters are not permitted for the BUSY command (1 byte command).

Abbreviation: IDENT
Comm. code: [02h]
Description: Reading the WABCO data.
Example: [02h] [EXECUTE]
Remark: The ECU sends to the tester
[<81+n>h] [02h] [<data 1>]...[<data n>] [EXECUTE]

Abbreviation: RESET
Comm. code: [03h]
Description: Electronic unit set to a defined initial state.
Example: [03h] [EXECUTE] or [<81+n>h] [03h] [<data 1>]...[<data n>] [EXECUTE]
Remark: The ECU sends an ACK command (acknowledgement) to the tester.

Abbreviation: ACK
Comm. code: [04h]
Description: Response to a command or data transmission
Example: [04h] [EXECUTE] or [<81+n>h] [04h] [<data 1>]...[<data n>] [EXECUTE]
Remark: Specification by parameters is allowed.

Abbreviation: ERROR
Comm. code: [05h]
Description: Error message
Example: [05h] [EXECUTE] or [<81+n>h] [05h] [<data 1>]...[<data n>] [EXECUTE]
Remark: The error may be specified by parameters.

Abbreviation: WR_PASS
Comm. code: [06h]
Description: Transmission of the pass word to the electronic unit in order to switch over to the supervisor mode.
Example: [82h] [06h] [<Passwort>] [EXECUTE] or
[<81+n>h] [06h] [<data_1>]...[<data_n>] [EXECUTE]
Remark: The electronic unit acknowledges the supervisor mode by sending a ACK command, and a faulty pass word by sending an ERROR command.

Abbreviation: RD_PASS
Comm. code: [07h]
Description: Request for the random data for the pass word in order to switch over to the supervisor mode.
Example: [07h] [EXECUTE]
Remark: The ECU responds:
[82h] [07h] [<random data>] [EXECUTE] or
[<81+n>h] [07h] [<data_1>]...[<data_n>] [EXECUTE]

Abbreviation: READ_KENNUNG
Comm. code: [08h]
Description: Output of the diagnostic code
Example: [08h] [EXECUTE]
Remark: The ECU responds:
[85h] [08h] [<parameter number>] [<diagnostic version number>]...
[<system identification>] [<system>] [EXECUTE].

Abbreviation: CHG_CPU
Comm. code: [0Eh]
Description: Switching the CPU in multi-processor systems.
Example: [0Eh] [EXECUTE] or [82h] [0Eh] [<CPU No.>] [EXECUTE]
Remark: The selected CPU responds with an ACK command.

Abbreviation: EXIT
Comm. code: [0Fh]
Description: The ECU leaves the service mode.
Example: [0Fh] [EXECUTE]
Remark: The unidirectional free-running diagnosis is started after $t \geq 20$ ms or after the ignition has been switched off and on again (cf. 4.4).

5.4.5 Description of the standard functions

Abbreviation: CLR_FSP
Comm. code: [10h]
Description: Clearing the error memory
Example: [10h] [EXECUTE] or
[<81+n>h] [10h] [<data 1>]...[<data n>] [EXECUTE]
Remark: The ECU acknowledges the clearing of the error memory by sending an ACK command to the tester.

Abbreviation: RD_FSP
Comm. code: [11h]
Description: Reading the error memory
Example: [11h] [EXECUTE] or
[<81+n>h] [11h] [<data 1>]...[<data n>] [EXECUTE]
Remark: The read data are sent to the tester as follows: [<81+n>h] [11h] [<data 1>]... [<data n>] [EXECUTE].

Abbreviation: CLR_EEPROM
Comm. code: [12h]
Description: Clearing of data in the EEPROM.
Example: [82h] [12h] [<word address>] [EXECUTE] or
[83h] [12h] [<word address>] [<number of words>] [EXECUTE]
Remark: The ECU responds with an ACK command when the memory is cleared. Unrestricted access to the EEPROM is only allowed in the supervisor mode

Abbreviation: WR_EEPROM
Comm. code: [13h]
Description: Writing of data into the EEPROM.
Example: [84h] [13h] [<word address>] [<LSB data>] [<MSB data>] [EXECUTE]
Remark: The ECU responds with an ACK command when the data are stored in the memory.

Abbreviation: RD_EEPROM
Comm. code: [14h]
Description: Reading data from the EEPROM.
Example: [82h] [14h] [<word address>] [EXECUTE]
Remark: The ECU sends the data to the tester as follows:
[83h] [14h] [<LSB-Data>]...[<MSB data>] [EXECUTE]. Unrestricted access to the EEPROM is only allowed in the supervisor mode.

Abbreviation: WR_SP
Comm. code: [15h]
Description: Storage of data.
Example: [83h] [15h] [<address>] [<data>] [EXECUTE] or
[<83+n>h] [15h] [<LSB address>] [<MSB address>] [<data 1>]... [<data n>] [EXECUTE]
Remark: Only in the supervisor mode. The ECU sends an ACK command after execution of the command.

Abbreviation: RD_SP
Comm. code: [16h]
Description: Reading data.
Example: [82h] [16h] [<address>] [EXECUTE] or
[84h] [16h] [<LSB address>] [<MSB address>] [<number of bytes>] EXECUTE]
Remark: Only in the supervisor mode. The ECU sends the data to the tester as follows: [<81+n>h] [16h] [<data 1>]...[<data n>] [EXECUTE].

5.4.6 Description of the system commands

Abbreviation:	DO_xx
Comm. code:	[<20+xx>h]
Description:	Execution of the function xx.
Example:	[<20+xx>h] [EXECUTE] or [<81+n>h] [<20+xx>h] [<data 1>]...[<data n>] [EXECUTE]
Remark:	xx = {00h - 1Fh}. If the ECU does not find any data to be tranmitted to the tester, the ECU acknowledges the execution of the function with an ACK command. Otherwise the first data byte must be transmitted before time-out 0 has been reached in the tester (cf. 5.3). Relevant provisions shall be given in the Product Specification.

Abbreviation:	RD_xx
Comm. code:	[<40+xx>h]
Description:	Reading the value or the values xx.
Example:	[<40+xx>h] [EXECUTE] or [<81+n>h] [<40+xx>h] [<data 1>]...[<data n>] [EXECUTE]
Remark:	xx = {00h - 1Fh}; The ECU responds: [<81+n>h] [<40+xx>h] [<data 1>]...[<data n>] [EXECUTE]. Relevant provisions shall be given in the Product Specification.

Abbreviation:	SYSPRG1_xx
Comm. code:	[<60+xx>h]
Description:	Start of the system-specific program xx.
Example:	[<60+xx>h] [EXECUTE] or [<81+n>h] [<60+xx>h] [<data 1>]...[<data n>] [EXECUTE]
Remark:	xx = {00h - 0Fh}. If the ECU does not find any data to be tranmitted to the tester, the ECU acknowledges the execution of the function with an ACK command. Otherwise the first data byte must be transmitted before time-out 0 has been reached in the tester (cf. 5.3). Relevant provisions shall be given in the Product Specification.

Abbreviation:	SYSPRG2_xx
Comm. code:	[<70+xx>h]
Description:	Start of the system-specific program xx.
Example:	[<70+xx>h] [EXECUTE] or [<81+n>h] [<70+xx>h] [<data 1>]...[<data n>] [EXECUTE]
Remark:	xx = {00h - 0Fh}; the tester does not send further commands until time-out 1 has expired. During this period the ECU can send data to the tester without handshake (see time-out 1). The first byte must include the number of the following bytes. Acknowledgement with handshake is possible after the next NOP command from the tester. Relevant provisions shall be given in the Product Specification.

6 Identification parameters of the electronic unit

The structure of the EEPROM or ROM has to be specified in the Product Specification for each electronic unit.

6.1 Manufacturing data

Six words shall be reserved in the EEPROM or ROM for WABCO data. Their addresses shall be provided in the Product Specification. Appendix A shows the description of the WABCO data.

For the recognition of errors a 16 bit checksum is used. The checksum for WABCO data is calculated by adding the unsigned data byte by byte.

Checksum = $\sum_{i=0}^{i=9} |Xi| + 1$ X = { 8 bit data byte }

WABCO Data			
Word	**Byte**	**Description**	
1	LSB	Key word 1	(see
	MSB	Key word 2	Appendix D)
2	LSB	WABCO No. 1	
	MSB	WABCO No. 2	
3	LSB	WABCO No. 3	
	MSB	Software version number	
4	LSB	Week of production	
	MSB	Year of production	
5	LSB	Status byte	
	MSB	Reserved	
6	LSB	16 bit checksum for	
	MSB	WABCO data	

6.2 Diagnostic code

The diagnostic code determines the diagnostic software to be used by the tester and is output through the command "Read_Kennung".

Four bytes shall be reserved in the EEPROM or ROM for the diagnostic code.

Byte	Description
1	Parameter version
2	Diagnostic software version
3	System variant
4	System

The data codes shall be provided for each system in the relevant Product Specification. The respective address shown in Appendix C shall be stored in the parameter "System".

6.3 Tester identification

In case of access to a protected memory section the electronic unit stores a set of data for later identification of the tester.

Byte	Description
1	ID code
2	Company code
3	LSB tester number [1)]
4	MSB tester number

[1)] WABCO card number or customer code

ID Code	Description
1111 1111b:	EEPROM cleared
0000 0000b:	Production code
0000 1111b:	WABCO card number
1111 0000b:	Customer-specific code

When the random value for the calculation of the tester identification is requested, the tester transmits the data to be stored with the command RD_PASS.

Tester request:	[84h] [RD_PASS] [<comany code>] [<LSB tester number>] ... [<MSB tester number>] [EXECUTE]
ECU response:	[<81+n>h] [RD_PASS] [<data 1>].....[<data n>] [EXECUTE] Data: byte 1 to n of the random value

The electronic unit determines the ID code by means of the company code and the tester identification to be calculated. In order to prevent the faulty use of company codes, a secret company identification is allocated to each company code. The company identification is taken into account when the tester identification is calculated on the basis of an algorithm which is to be determined individually for each ECU.

Die electronic unit stores the data when the tester subsequently transmits the correct tester identification using the command WR_PASS, clearing the access to the protected section.

Tester request:	[<81+n>h] [WD_PASS] [<data 1>].....[<data n>] [EXECUTE] Data: byte 1 to n of the tester identification
ECU response:	[81h] [ACK] [EXECUTE] ⇨ correct, set of data stored or [81h] [ERROR] [EXECUTE] ⇨ incorrect, set of data not stored

The previously stored set of data is overwritten so that the last access is always recorded. The tester can only read the set of data.

Appendix H represents the procedure.

Pietsch

Appendix A: Description of the WABCO data in the EEPROM

word	byte	Description	Code
1	LSB	Key word 1 (low byte)	7 0 Hex: Parity bit; odd parity
	MSB	Key word 2 (high byte)	7 0 Hex: Parity bit; odd parity
2	LSB	WABCO Product Number	7 0 Hex: 884 446 7 0
	MSB		7 0 Hex: 5 0 9 8
3	LSB		7 0 Hex: 3 0 9 6
	MSB	Software version number	7 0 Hex:
4	LSB	Date of production	7 0 Hex: 7 6 5 0 Week of product. 1 - 52 Century of prod. 00 = 19 01 = 20
	MSB		7 0 Hex: 7 6 0 Year of prod. 0 - 99 Prototype 0 = 884...... Series unit 1 = 446......
5	LSB	Status byte	7 0 Hex: 7 1 0 0: WABCO daten written in 1: EEPROM cleared Not assigned
	MSB	Reserved for future use	7 0 Hex:
6	LSB	Checksum for WABCO data	7 0 Hex: 7 0
	MSB		7 0 Hex: 15 8 Checksum = $\sum_{i=0}^{i=9} \lvert X_i \rvert + 1$ X = {8-bit data}

Appendix B: Free-running Protocol

|<----- ISO 9141 ----->|

T0	SYN	T1	KEY1	T2	KEY2	T3	WNR1	WNR2	WNR3	SNR	DAT1	DAT2	STS	CCE	NFM	FM1		NMW	MW1		MWn

|<------------------------------------ MAX. 0,5s ------------------------------------>|

ABBREVIATION	DESCRIPTION	BYTE FORMAT
T0 $\geq$ 20ms	FRAME SYNCHRONIZATION TIME	
SYN	BAUD RATE SYNCHRONIZATION BYTE	START BIT / 8 ALTERNATING BITS/ STOP BIT
2ms $\leq$ T1 < 20ms	BAUD RATE SYNCHRONIZATION TIME	
KEY1	KEY WORD 1 (see Appendix D)	START BIT / 7 DATA BITS / PARITY BIT (ODD) / STOP BIT
0,2ms $\leq$ T2 < 20ms	TIME BETWEEN THE KEY WORDS	
KEY2	KEY WORD 2 (see Appendix D)	
2ms $\leq$ T3 < 20ms	TIME FOR SELECTION OF TEST SOFTWARE	
WNR1	1. BYTE WABCO PRODUCT NUMBER	START BIT / 8 DATA BITS / STOP BIT
WNR2	2. BYTE WABCO PRODUCT NUMBER	
WNR3	3. BYTE WABCO PRODUCT NUMBER	
SNR	SOFTWARE VERSION NUMBER	
DAT1	1. BYTE DATE OF PRODUCTION	
DAT2	2. BYTE DATE OF PRODUCTION	
STS	STATUS BYTE	
CCE	INSTANTANEOUS ERROR CODE/CODE NUMBER	
NFM	NUMBER OF FOLLOWING ERROR MEMORY BYTES	
FM1...FMn	ERROR MEMORY BYTES	
NMW	NUMBER OF FOLLOWING MEASURING VALUES	
MW1...MWn	MEASURING VALUES	" BYTES FOR EACH MEAS.VALUE (LSB FIRST)

TIME T4 BETWEEN THE TRANSMISSION OF TWO CONSECUTIVE DATA BYTES 0,2ms $\leq$ T4 < 20ms

Appendix C: Assignment of the 5 baud addresses

Based on SAE 1708 the following addresses are recommended:

Address	Electronic system
00 – 07	Engine
 08 09	Brakes, tractor ABS Available for use
 10 11	Brakes, trailer ABS Available for use
12 – 13	Tires, tractor
14 – 15	Tires, trailer
 16 17	Suspension, tractor ECAS Available for use
 18 19	Suspension, trailer ECAS Available for use
 20 21 – 27	Transmission EPS Available for use
20 – 32	Electrical
33 – 35	Cargo refrigeration/heating
36 – 40	Instrument cluster
 41 42 – 45	Driver information Driver information center Available for use
 46 47	Cab climate control ATC Available for use
48 – 55	Reserved
56 – 61	Trip recorder
62 – 63	Turbocharger
64 – 68	Reserved
69 – 111	Reserved for interfaces
112	Speed limiter
113 – 127	Available for use

Appendix D: Key word assignment

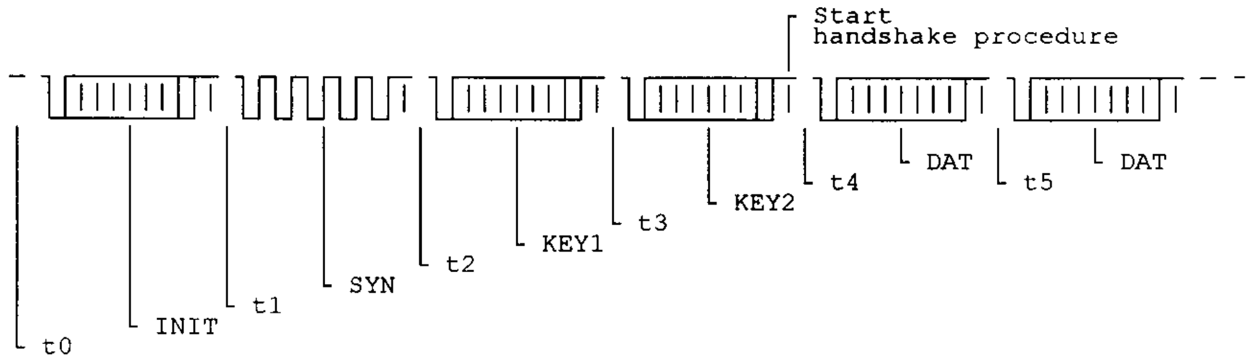
The key words only specify the subsequently used communication protocol, but not the electronic unit. Therefore the same key words are assigned for all electronic units communicating with the tester according to the same protocol. The key words are changed only if the used protocol is changed or modified.

The table below shows the key words assigned to WABCO and the allocated protocols.

	Key word I	Key word II	Protocol
Dez.: Hex.: ISO :	121 79 79H	0 00 80H	Diagnostic concept to JED-677
Dez.: Hex.: ISO :	122 7A 7AH	0 00 80H	
Dez.: Hex.: ISO :	123 7B FBH	0 00 80H	
Dez.: Hex.: ISO :	124 7C 7CH	0 00 80H	
Dez.: Hex.: ISO :	125 7D FDH	0 00 80H	

Remark: The "ISO" key words include the parity bit (MSB; odd parity) that is specified in ISO 9141.

Appendix E: Initialization of an ECU for bidirectional diagnosis



2ms ≤ t0 ≤ ∞

INIT: Initialization of an ECU for bidirectional diagnosis.
- 5 baud address from the tester
  (start bit "0" / 7 data bits / odd parity bit / stop bit "1")
- logic "0" for 1,8 s ± 0,01 s through tester
- short to ground for t ≥ 2 s

Remark: An ECU recognizes the initialization signal on lines K or L or on both lines.

2 ms ≤ t1 ≤ 2 s
SYN: Baud rate synchronization byte from the ECU
(start bit "0" / 8 alternating bits (first bit "1") / stop bit "1")

2 ms*) ≤ t2 ≤ 1,2 s *) or for one bit time, depending on which time is longer

KEY1: Key word 1 (see Appendix D)
KEY1 and KEY2: specification of the subsequent serial data exchange and of the hardware configuration
(start bit "0" / 7 data bits / odd parity bit / stop bit "1")

0,2 ms ≤ t3 ≤ 1,2 s

KEY2: Key word 2 (see Appendix D)
(start bit "0" / 7 data bits / odd parity bit / stop bit "1").

Two further key words follow if all data bits of KEY1 and KEY2 are "1".

2 ms ≤ t4 < T1 |1|

DAT: Data
(1 digit: start bit "0" / 8 data bits / stop bit "1")

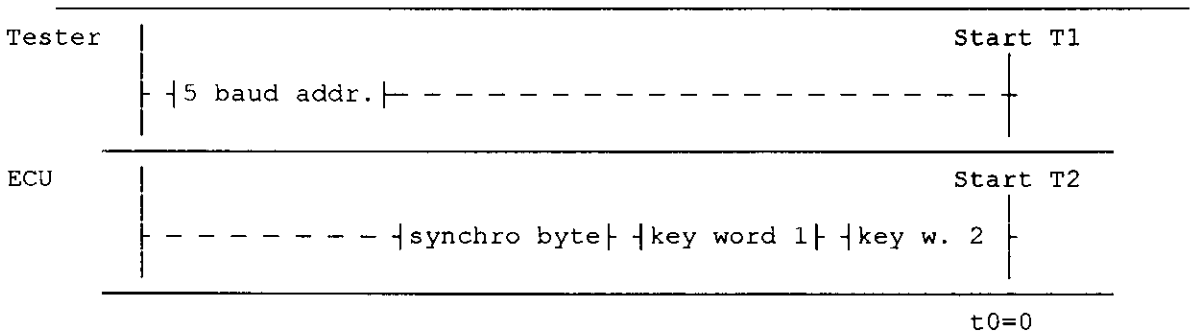
0,2 ms ≤ t5 < T0 (T1) |1|

The transmission of the data bits always starts with the LSB.

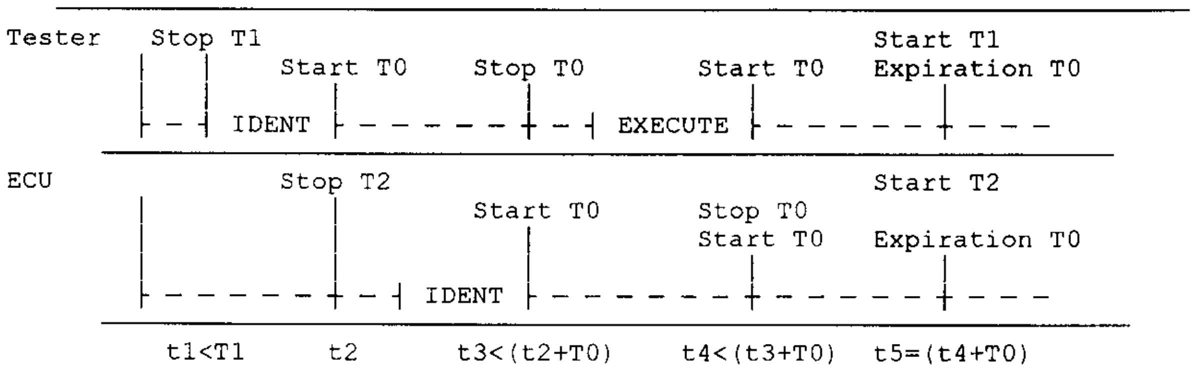
|1| T0 and T1 see section 5.3 "Data transmission and data safety"

Appendix F: Examples of the handshake procedure

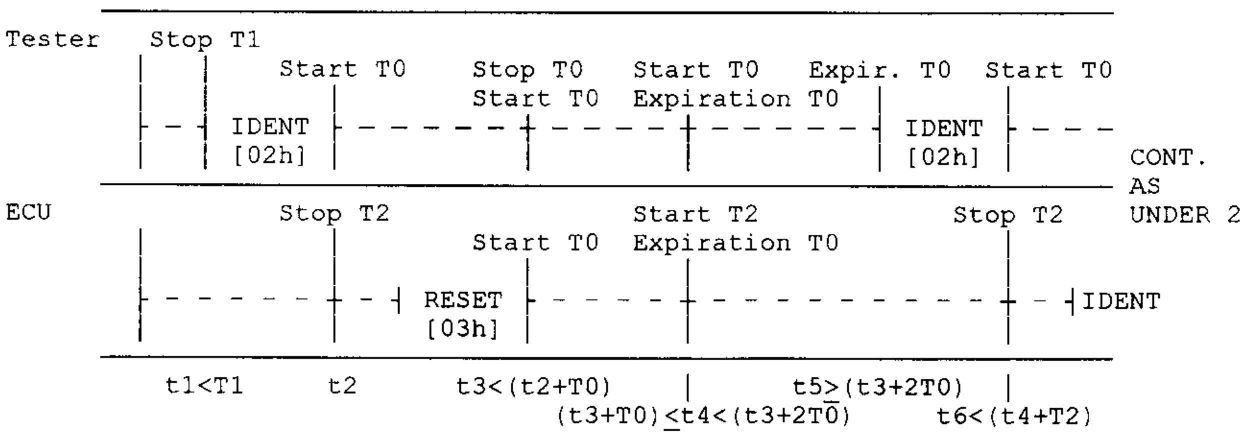
1 Start of the handshake procedure after initialization



2 Faultless transmission of the IDENT command from the tester to the ECU



3 Faulty transmission of the IDENT command from the tester to the ECU



Continuation of Appendix F

4 Faultless transmission of the ACK command from the ECU to the tester

```
ECU      Stop T2                                                Start T2
         |   |    Start T0     Stop T0       Start T0   Expiration T0
         |   |                |          |                |           |
         |- -|   ACK    |------------+ - -| EXECUTE |-----------+---
Tester             Stop T1                                      Start T1
         |              |      Start T0      Stop T0
         |              |              |            Start T0   Expiration T0
         |              |              |                 |           |
         |-------------+ - -| ACK |---------------+-----------+---
              t1       t2<T1    t3<(t2+T0)     t4<(t3+T0)   t5=(t4+T0)
```

5 Faulty transmission of the ACK command from the ECU to the tester

```
ECU      Stop T2
         |   |    Start T0     Stop T0      Start T0      Expir.T0  Start T0
         |   |                 Start T0     Expiration T0  |          |
         |- -|   ACK    |------------+-----------+--------------|  ACK   |- - -
         |   |  [04h]   |            |           |              | [04h]  |       CONT.
Tester             Stop T1                  Start T1                Stop T1      AS
         |              |      Start T0     Expiration T0              |         UNDER 4
         |              |             |                                |
         |-------------+ - -| ERROR |----------+--------------------------+ - -|IDENT
         |              |    [05h]   |          |                         |
             t1      t2<T1     t3<(t2+T0)  |          t5≥(t3+2T0)   |
                                   (t3+T0)≤t4<(t3+2T0)       t6<(t4+T1)
```

Appendix G: Examples of data transmission without handshake

1 Transmission of the command SYSPRG2 using handshake from the tester to the ECU

```
Tester   Stop T1                                     Start T1
         |   |     Start T0    Stop T0      Start T0   Expiration T0
         |   |         |          |            |           |
         |- -|SYSPRG2 |- - - - - -|- -| EXECUTE |- - - - - -|- - -

ECU              Stop T2                                Start T2
         |           |         Start T0     Stop T0
         |           |            |         Start T0   Expiration T0
         |           |            |            |           |
         |- - - - - -|- -|SYSPRG2 |- - - - - - -|- - - - - -|- -

           t1<T1     t2      t3<(t2+T0)   t4<(t3+T0)  t5=(t4+T0)
```

1.1 Maintaining the transmission status by transmitting BUSY commands using handshake from the ECU to the tester.

```
ECU      Stop T2                                     Start T2
         |   |     Start T0    Stop T0      Start T0   Expiration T0
         |   |         |          |            |           |
         |- -|  BUSY   |- - - - - -|- -| EXECUTE |- - - - - -|- - -

Tester           Stop T1                                Start T1
         |           |         Start T0     Stop T0
         |           |            |         Start T0   Expiration T0
         |           |            |            |           |
         |- - - - - -|- -| BUSY   |- - - - - - -|- - - - - -|- -

            t6      t7<T1    t8<(t7+T0)   t9<(t8+T0)  t10=(t9+T0)
```

1.2 Transmission of data without handshake

```
ECU
         |
         |-|NUMB.BYTES|-|BYTE 1|-| BYTE 2|- - - - -| BYTE n|- - -
         |  [<80+n>h]                                    |

Tester
         | |                                             |
         |- - - - - - - - - - - - - - - - - - - - - - - - - - - - -
         | |                                             |

           t6                                          T7<(t5+T1)
   or      t11                                         t12<(t10+T1)
```

Appendix H: Device identification testing procedure