FIAT STANDARD DIAGNOSTIC PROTOCOL ON K–LINE KWP 2000

SUPERVISOR: Eng. ANTONIOLI Bruno – D.T. – F.V. – S.I.E.E. – S.S.E. – tel. 683.4766

MANAGER: Mr. CIRIMBILLI Riccardo – D.T. – F.V. – S.I.E.E. – S.S.E. – tel. 683.5423

PURPOSE

This document describes Fiat requirements for the implementation of Keyword Protocol 2000 (KWP2000) services on K–line, defining the general rules.

Change	Date	Description
=	Apr. '97	Edition 1 – New. (RG)
=	July '98	Edition 2 – Completely revised. (RG)
=	Mar. '99	Edition 3 – Completely revised. (RG)
A	Mar. '99	" " – Note at § 4.3 corrected. RLI list at § 7.1.2.1 modified. Some editing changes made (RG)
=	06.07.'00	Edition 4 – Following para. have been revised: • Para. 2.1 Standard ISO/Para. 2.2. Reference standards list updated • para. 4.3 Timing data changed • para. 4.4.3.3 Timing DownLoad Session • para. 4.5. communication start/stop conditions changed • para. 5.2.2. ResponseCodes table modified. (11h and 10h note NACK 0x78). • para. 6.1.2.1. diagnostic session added for starting of components 89h and 84 • para. 6.1.2.3. DiagnosticService List • para. 6.3 securityAccess service • para. 7.1.2.1. RLI 80h–9Fh management changed • 8.2.2.3. Error Memory structure • Para 12 ScanTool added

PUBLISHED BY SATIZ – NORMAZIONE

CONTENTS

continued

continued **CONTENTS**

continued

continued

CONTENTS

1

SCOPE

This document describes FIAT requirements for the implementation of the KWP 2000 communication protocol in FIAT products, defining the general rules; a document will be written for each application, containing the detailed description of each service. This document shall be named “Specifica di Diagnosi Finalizzata SDF xxxx”.

Said document will not contain the definition of the protocol toward the SCAN TOOL (ISO 14230−4).

For security reasons some implementation parts will not be described in any document but will be given only to persons responsible for such operations.

2

REFERENCE STANDARDS

2.1

ISO Standards

ISO 14230−1	Road Vehicles – Diagnostic Systems – Keyword Protocol 2000 Part 1 : Physical Layer
ISO 14230−2	Road Vehicles – Diagnostic SystemsKeyword Protocol 2000 Part 2 : Data Link Layer, Error Handling
ISO 14230−3	Road Vehicles – Diagnostic Systems – Keyword Protocol 2000 Part 1 : Implementation
ISO 9141−2	CARB requirements for interchange of Digital Information

2.2

Other Standards

FIAT 07234	General specifications for Self−diagnosis
SAE J 2012	Diagnostic Trouble Code
SAE J 2186	E/E Diagnostic Data Link Security
Keyword Protocol 2000	Implementation of Diagnostic services, Recommended Practice – version 1.4, September 16, 1997.

3

DEFINITIONS AND ABBREVIATIONS

3.1

Service identifiers

English	Italian
startCommunication	Inizio comunicazione
stopCommunication	Termine comunicazione
accessTimingParameter	Accesso ai parametri di comunicazione
startDiagnosticSession	Inizio sessione diagnostica
stopDiagnosticSession	Termine sessione diagnostica
securityAccess	Accesso alle procedure di sicurezza
testerPresent	Blocco di mantenimento comunicazione
readECUIdentification	Lettura codice identificativo ECU
readDataByLocalIdentifier	Lettura parametri
writeDataByLocalIdentifier	Scrittura parametri
readDiagnosticTroubleCodesByStatus	Lettura codici errori presenti in centralina raggruppati tramite il loro stato
readStatusOfDiagnosticTroubleCodes	Lettura parametri ambientali collegati ad un singolo errore
readFreezeFrameData	Lettura blocco informazioni catturate durante il rilevamento di anomalie rilevanti per le emissioni
clearDiagnosticInformation	Cancellazione memoria errori
inputOutputControlByLocalIdentifier	Operazioni effettuate su attuatori (attivazioni, tarature, etc.)
startRoutineByLocalIdentifier	Attivazione programma residente
stopRoutineByLocalIdentifier	Interruzione programma residente attivato in precedenza
requestRoutineResulByLocalIdentifier	Informazioni sullo stato programma (in esecuzione) attivato in precedenza
requestDownload	Richiesta di programmazione
transferData	Comando utilizzato per trasferire i dati da programmare
requestTransferExit	Richiesta chiusura procedura di programmazione

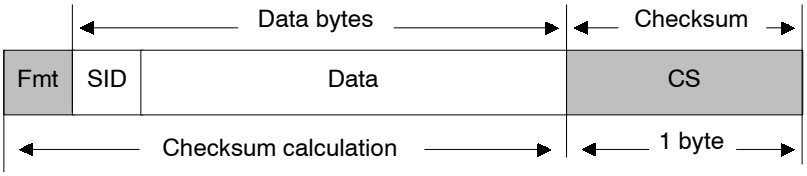
4

COMMUNICATION

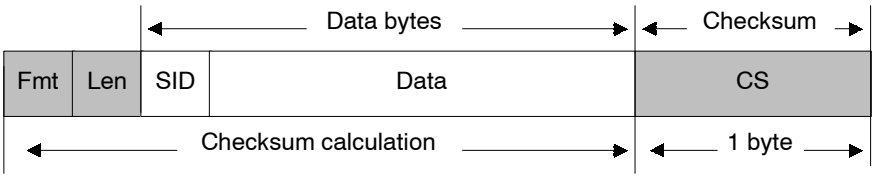
4.1

Introduction

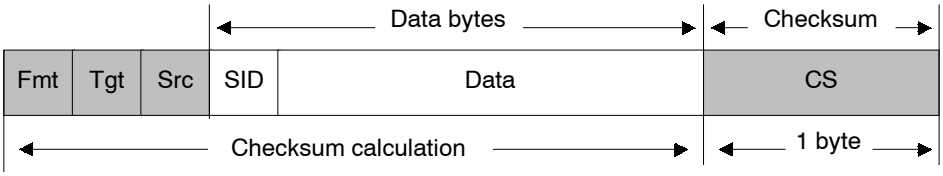
The structure of the messages is described in the document ISO 14230−2; in the following paragraph are briefly described the possible formats and some limitations/differences with respect to the reference document.



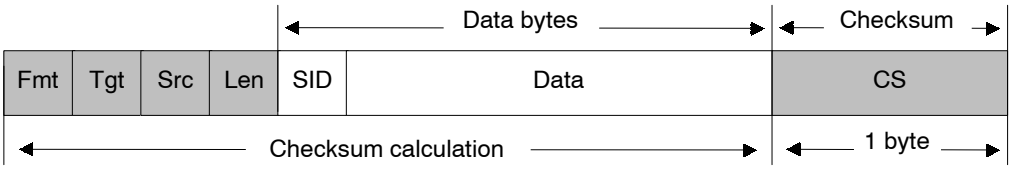
Header without address information, no additional lenght byte



Header without address Information, additional length byte



Header with address Information, no additional length byte



Header with address Information, 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)		

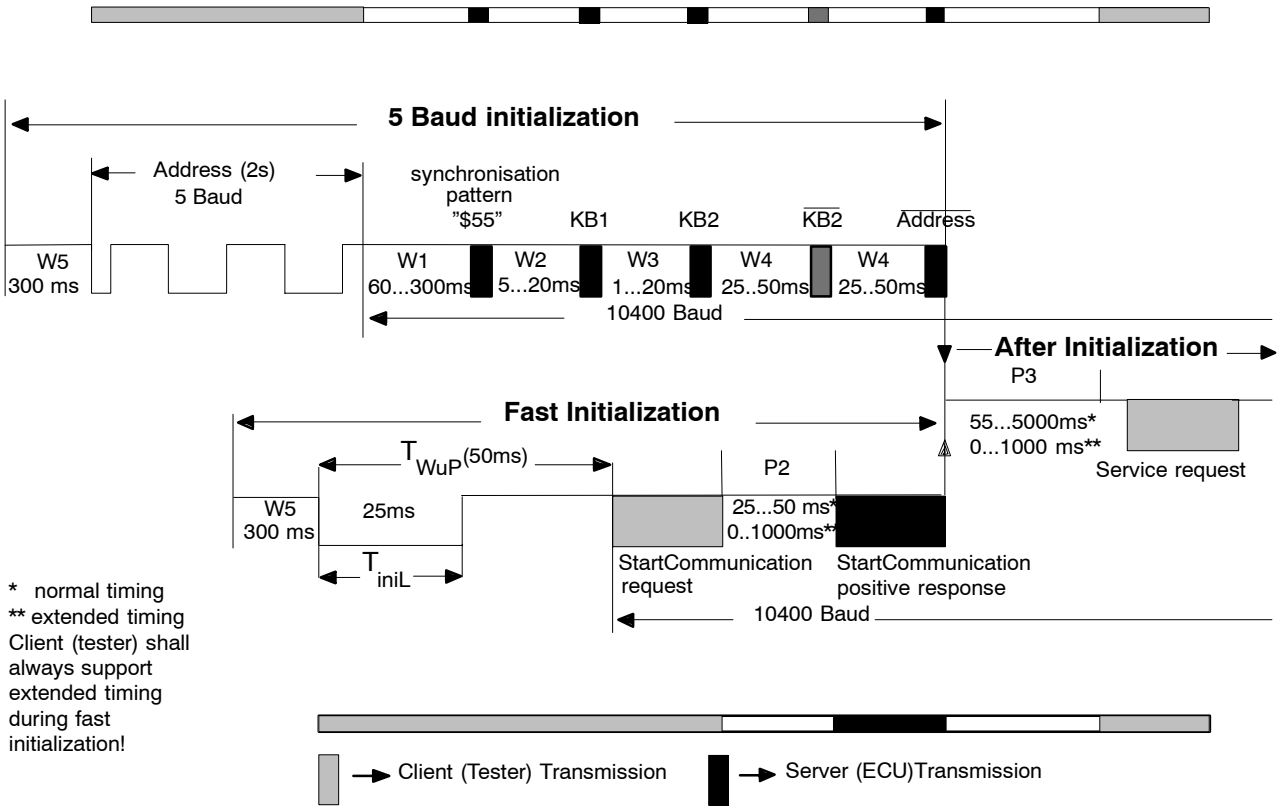
The data field is described in this document. Shaded areas (header, checksum) are described in ”Keyword Protocol 2000 – Part 2: Data Link Layer

N.B.: For the response block the ECU shall always use the same format sent by the Tester for the request block.

4.2

Initialization

The ECU shall accept both 5 Baud initialization and Fast Initialization.



4.2.1

Fast initialisation

See Std. ISO 14230−2

4.2.2

5−Bd initialization

After receiving the 5−baud address, the ECU transmits the synchronization sequence “55H” and two key bytes **(*)**. The Tester shall transmit the complement of key byte 2 after which the ECU will transmit complement of the address.

If the ECU doesn’t get this byte correctly it shall stop communication and automatically reset for a new diagnostic session. After a waiting time w_5 the Tester is allowed to retry initialization.

(*) The baud rate used by both the E.C.U. and Tester shall be **10.4 Kb ±1%**.

4.2.2.1

Initialization timings diagram

The initialization has the following timing diagram:

– Time before initialization : $w_5 > 300ms$.

– Time between initialization and ISO code : $60 ms < w_1 < 300ms$.

– Time between 1st byte ISO code and Key 1 : $5 ms < w_2 < 20 ms$

– Time between Key 1 and Key 2 : $1 ms < w_3 < 20 ms$

– Time between complement of Key 1 and complement of address : $25 ms < w_4 < 50 ms$

4.2.3

Key Bytes

Through the two Key Bytes the Tester is informed about which communication timings and formats are supported by the ECU.

The same Key Bytes shall be assigned respecting Std. ISO 14230–2; the criteria on how to set the two Key Bytes so as to support ECU contents are given hereafter.
The ECU shall use the same kind of Header used by the Tester.

Keybytes			Supported		
Bynary		Hex	Length	Type of header	
KB2	KB1		information		
1000 1111	1101 0000	8FD0h	See note		
1000 1111	1101 0101	8FD5h	format byte	1 byte header	
1000 1111	1101 0110	8FD6h	additional length byte		
1000 1111	0101 0111	8F57h	both modes possible		
1000 1111	1101 1001	8FD9h	format byte	Header target and source address information	Extended timing
1000 1111	1101 1010	8FDAh	additional length byte		
1000 1111	0101 1011	8F5Bh	both modes possible		
1000 1111	0101 1101	8F5Dh	format byte	Both types of header supported	
1000 1111	0101 1110	8F5Eh	additional length byte		
1000 1111	1101 1111	8FDFh	both modes possible		
1000 1111	1110 0101	8FE5h	format byte	1 byte header	
1000 1111	1110 0110	8FE6h	additional length byte		
1000 1111	0110 0111	8F67h	both modes possible		
1000 1111	1110 1001	8FE9h	format byte	Header target and source address information	Normal timing
1000 1111	1110 1010	8FEAh	additional length byte		
1000 1111	0110 1011	8F6Bh	both modes possible		
1000 1111	0110 1101	8F6Dh	format byte	Both types of header supported	
1000 1111	0110 1110	8F6Eh	additional length byte		
1000 1111	1110 1111	8FEFh	both modes possible		

NOTE: With value 8FD0H the ECU does not provide any information about which option of the standard is supported.

These options refer to the use of normal or extended timing, additional length byte, header with or without address information.

In case of 5–Baud initialization the Tester has to know which options are implemented.

4.2.4

Physical addresses assigned to ECU and Tester

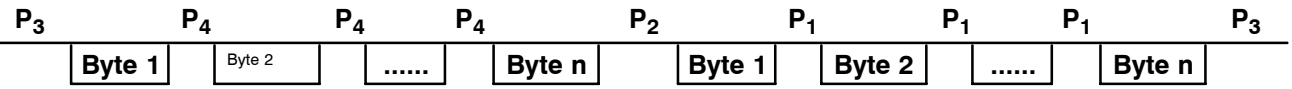
For the request message transmitted by Tester the Target Address is always the physical address of the ECU; for the ECU the Target Address in the response message is always the physical address of the Tester.

Addresses are defined by FIAT.

ECU Physical Address (Hex)	Tester Physical Address (Hex)
TBD	F1

4.3

Communication timings



Block 1 (Tester) **Block 2 (ECU)**

– ECU interbyte time : **2 ms < P_1 < 20 ms**

– Time between Tester request and subsequent ECU response or between two consecutive ECU responses : **25 ms < P_2 < 50 ms**

– Time between ECU response and new Tester request : **55 ms < P_3 < 5000 ms**

– Tester interbyte time : **5 ms < P_4 < 20 ms**

NOTE: In case of reprogramming (Diagnostic Session 0x85) or in case of not valid Flash ROM data, the control unit can use a **P_1 = 0ms**.

4.4

Communication services

The following chapters contain a detailed description of the Data field for all communication services; all other fields are defined in document ISO 14230−2.

4.4.1

startCommunication service

4.4.1.1

Message data bytes

startCommunication Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	startCommunication Request service Id	M	81

startCommunication Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	startCommunication Positive Response service Id	S	C1
#2	Key Byte #1	M	See 4.2.3
#3	Key Byte #2	M	

No negative response is contemplated.

4.4.2

stopCommunication service

4.4.2.1

Message data bytes

stopCommunication Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	stopCommunication Request service Id	M	82

stopCommunication Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	stopCommunication Positive Response service Id	S	C2

stopCommunication Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	stopCommunication Request service Id	M	82
#3	responseCode = [see § 5.2.2 − Parameter Definitions]	M	xx

4.4.3

accessTimingParameter service

4.4.3.1

Message data bytes

accessTimingParameter Positive Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	AccessTimingParameter Request service Id	M	83
#2	Communication Parameter Identifier – see § 4.4.3.2	S	xx
#3	P2 Min [0.5 ms/bit]	C	xx
#4	P2 Max [25 ms/bit ;FFh = ∞]	C	xx
#5	P3 Min [0.5 ms/bit]	C	xx
#6	P3 Max [250 ms/bit;FFh = ∞]	C	xx
#7	P4 Min [0.5 ms/bit]	C	xx

C : parameters are present only if Communication Parameter Identifier (CPI) is equal to 03.

accessTimingParameter Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	accessTimingParameter Request service Id	M	C3
#2	Communication Parameter Identifier – see § 4.4.3.2	S	xx
#3	P2 Min [0.5 ms/bit]	C	xx
#4	P2 Max [25 ms/bit ;FFh = ∞]	C	xx
#5	P3 Min [0.5 ms/bit]	C	xx
#6	P3 Max [250 ms/bit;FFh = ∞]	C	xx
#7	P4 Min [0.5 ms/bit]	C	xx

C : parameters are present only if CPI is equal to 00 or 02.

accessTimingParameter Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negative Response service Id	S	7F
#2	Communication Parameter Identifier – see § 4.4.3.2	M	83
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

4.4.3.2

Parameter Definitions

Data byte #	Value
00	Read possible limits
01	Reset to default values
02	Read current values
03	Set values

4.4.3.3

Timing DownLoad Session

The control unit must be designed for accepting the following limits during the diagnostic download session (0x85), besides the default values and its own physical limits:

0 <= P1 <= 10 ms
15 <= P2 <= 50 ms
0.5 <= P3 <= 5000 ms
0 <= P4 <= 20 ms

4.5

Protocol Structure

The communication can start only under the following conditions:

- +15 ON (from wire of via CAN) (1)

The communication must keep active through a data exchange of request/response type or through testerPresent service.

After one time−out, the ECU must get back to normal mode service. The communication must be restored by the Tester through a new startCommunication or 5 Baud, in the meantime, of course, each pending service must be aborted.

If the Tester sends a new request startDiagnosticSession while a Diagnosis Session is running (f.e. for changing diagnosticMode), the new session must start only after ECU has aborted all pending services of the current diagnostic session.

The diagnostic communication must finish when at least one of the following condition is met:

- +15 OFF (except for particular cases, f.e. Body Computer)
- a time−out in the communication occurs
- the Tester has required a stopCommunication to stop the default session which had been started during the communication initialization.

If during the starting of an ECU component (inputOutputControlByLocalIdentifier) the diagnosis session or the communication are stopped, the ECU must automatically have the total control of the component in order to avoid any possible damages.

(1) In case the signal +15 (ignition−on) is not used, the agreement between DT SIEE and system Supplier is valid, and they will be added to the Finalized Diagnosis Specification.

5

GENERAL RULES

5.1

Service Identifier

The following table lists all serviceIdentifier's provided in the FIAT implementation of KWP2000.

Name of Service	Request Hex Value	Response Hex Value
startCommunication	81	C1
stopCommunication	82	C2
accessComunicationParameter	83	C3
startDiagnosticSession	10	50
stopDiagnosticSession	20	60
securityAccess	27	67
testerPresent	3E	7E
readECUIdentification	1A	5A
readDataByLocalIdentifier	21	61
writeDataByLocalIdentifier	3B	7B
readDiagnosticTroubleCodesByStatus	18	58
readStatusOfDiagnosticTroubleCodes	17	57
readFreezeFrameData	12	52
clearDiagnosticInformation	14	54
inputOutputControlByLocalIdentifier	30	70
startRoutinebyLocalIdentifier	31	71
stopRoutineByLocalIdentifier	32	72
requestRoutineResultByLocalIdentifier	33	73
requestDownload	34	74
transferData	36	76
requestTransferExit	37	77

5.2

negativeResponse

5.2.1

Message data bytes

negativeResponse Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	Request service Id	M	xx
#3	responseCode	M	xx

5.2.2

Parameter Definitions

ResponseCode parameter values are detailed in /**9**/.

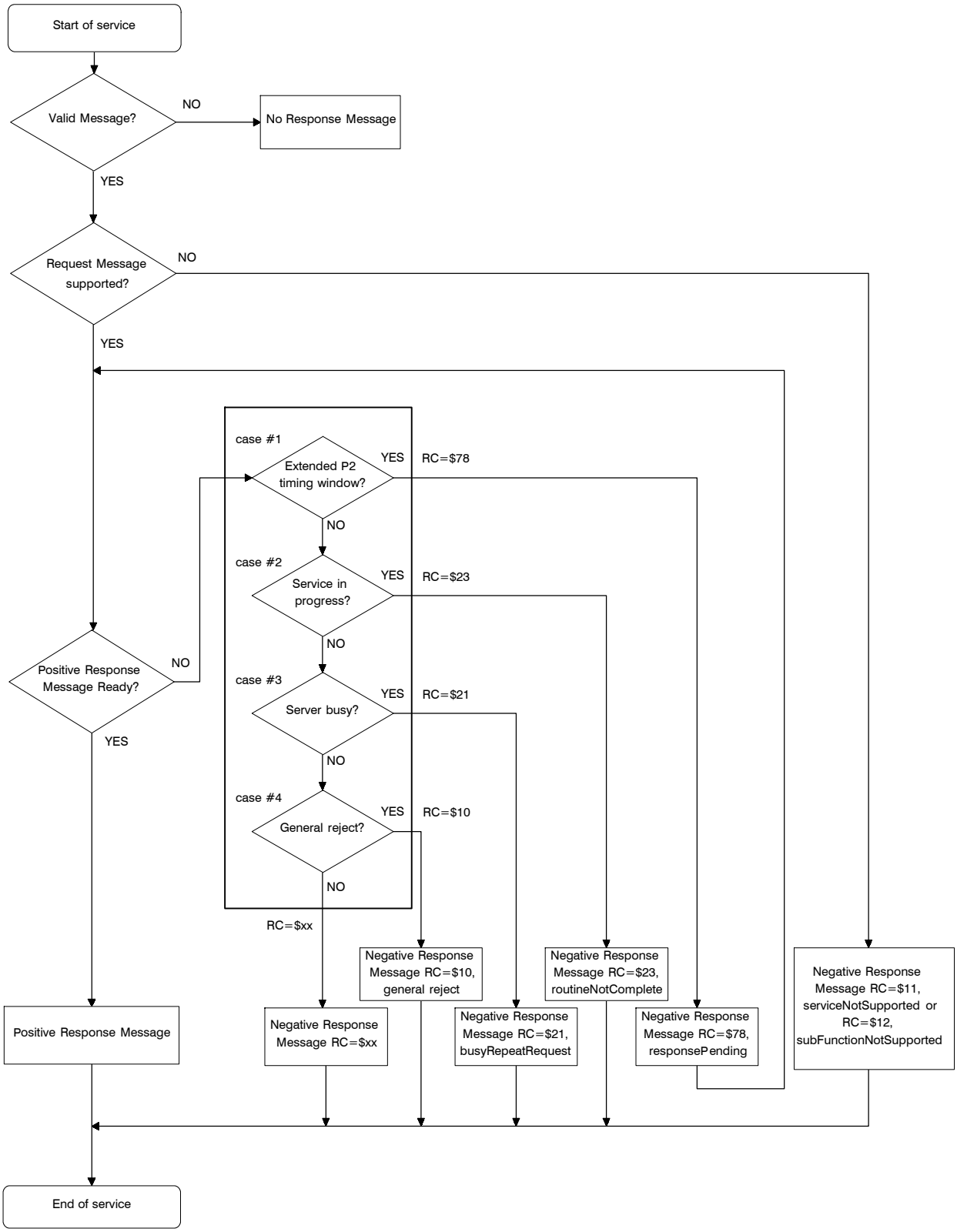
Sections 6−11 of said document contain a list of suggested values for responseCode to be used for all specified services. Additional values for the responseCode parameter shall be implemented by the ECU Supplier only after approval by FIAT. FIAT reserves the right to request specific responseCode values in the range vehicleManufacturerSpecific (90h−F9h) in case of necessity.

The following table lists responseCode's suggested by FIAT for each of the services implemented in the KWP2000 on K−line. Suppliers shall make use of this table to list all responseCode's actually implemented for each of the services provided by their ECU.

serviceIdentifiers / responseCodes	(82h) stopCommunication	(83h) accessTimingParameter	(10h) startDiagnosticSession	(20h) stopDiagnosticSession	(27h) securityAccess	(3Eh) testerPresent	(1Ah) readECUIdentification	(21h) readDataByLocalIdentifier	(3Bh) writeDataByLocalIdentifier	(18h) readDiagnosticTroubleCodesByStatus	(17h) readStatusOfDiagnosticTroubleCodes	(12h) readFreezeFrameData	(14h) clearDiagnosticInformation	(30h) InputOutputControlByLocalIdentifier	(31h) startRoutineByLocalIdentifier	(32h) stopRoutineByLocalIdentifier	(33h) requestRoutineResultByLocalIdentifier	(34h) requestDownload	(36h) transferData	(37h) requestTransferExit
(10h) generalReject																				
(11h) serviceNotSupported	$	$	$	$	$	$	$	$	$	$	$	$	$	$	$	$	$	$	$	$
(12h) subFunctionNotSupported–invalidFormat		#	#		#		#	#	#	#	#	#	#	#	#	#	#			
(21h) busy–repeatRequest									#				#				#			
(22h) conditionsNotCorrectOr RequestSequenceError			#		#				#				#	#	#	#	#	#		#
(23h) routineNotComplete													#				#			
(31h) requestOutOfRange		#	#						#						#		#			
(33h) securityAccessDenied–securityAccessRequested					#			#	#						#			#		
(35h) invalidKey					#															
(36h) exceedNumberOfAttempts					#															
(37h) requiredTimeDelayNotExpired					#															
(40h) downloadNotAccepted																		#		
(41h) improperDownloadType																		#		
(42h) canNotDownloadTo SpecifiedAddress																		#		
(43h) canNotDownloadNumberOf BytesRequested																		#		
(50h) uploadNotAccepted																				
(51h) improperUploadType																				
(52h) canNotUploadFrom SpecifiedAddress																				
(53h) canNotUploadNumberOf BytesRequested																				
(71h) transferSuspended																				
(72h) transferAborted																			#	#
(74h) illegalAddressInBlockTransfer																				
(75h) illegalByteCountInBlockTransfer																			#	
(76h) illegalBlockTransferType																			#	
(77h) blockTransferDataChecksumError																				
(78h) requestCorrectlyReceived–ResponsePending									#				#						#	#
(79h) incorrectlyByteCountDuring BlockTransfer																				
(80h) serviceNotSupportedInActive DiagnosticMode																				

Legend:
- ☐ responseCode not to be used for this serviceIdentifier
- # responseCode suggested for this serviceIdentifier
- √ responseCode used for this serviceIdentifier
- $ mandatory responseCode if the serviceIdentifier is not supported

Following flow−chart illustrates the general rules for the assignment of correct codes to the negativeR-esponse message:



Response codes different from 10h, 11h, 12h, 23h and 78h shall be assigned according to /**9**/.

The following tables contain some examples of negative response messages with response code 23h ("requestCorrectlyReceived−ResponseP ending") or 78h ("reqCorrectlyRcvd−RspP ending ").

time	client (Tester)	server (ECU)
P3	<Service Name> Request [...]	**{ server has started routine! }**
P2		<Service Name> NegativeResponse [routineNotComplete ($23)]
P3	<Service Name> Request [...]	
P2		<Service Name> NegativeResponse [busy−RepeatRequest ($21)]
	:	:
P3	<Service Name> Request [...]	
P2		<Service Name> NegativeResponse [busy−RepeatRequest ($21)]
P3	<Service Name> Request [...]	**{ server has stopped routine! }**
P2		<Service Name> PositiveResponse [...]
		OR
P2		<Service Name> NegativeResponse [RC ≠ busy−RepeatRequest ($21)]

time	client (Tester)	server (ECU)
P3	<Service Name> Request[...]	
P2		<Service Name>NegRsp#1 [reqCorrectlyRcvd−RspP ending ($78)]
		:
P2*		<Service Name>NegRsp#n [reqCorrectlyRcvd−RspP ending ($78)]
P2*		<Service Name> PositiveResponse [...]
P3	<Service Name> Other Request[...]	
P2		<Service Name>Other PositiveResponse [...]

P2* : **P2max is set to P3max**.

6

DIAGNOSTIC MANAGEMENT FUNCTIONAL UNIT

6.0

Introduction

Services specified in the following paragraphs shall be used to enable/disable and maintain active various diagnostic modes in the ECU. At each moment it is possible to have only one active session. Each session enables a specific group of KWP2000 services which shall be defined by FIAT and may be protected by a security access mechanism.

If the Tester requests to open an already active diagnostic session, the ECU shall send a positive response message.

Mode defaultMode−StandardDiagnosticMode−OBDIIMode (see **6.1.2 Parameter Definitions**) is mandatory for all ECU's. Some ECU's may implement additional diagnostic modes, according to their specific diagnostic services. In such cases the ECU shall observe the following rules:

- If the Tester requests the opening of a diagnostic mode which is not active at the moment the request is made (e.g.: to change diagnosticMode) the ECU shall start the new session only after any pending service in the current diagnostic session has been aborted.

- It shall be possible to insert max. two diagnostic sessions: when the ECU receives a stopDiagnosticSession request, it shall stop (if current conditions allow it) the active session and resume mode defaultMode−StandardDiagnosticMode−OBDIIMode.

Every time a new diagnostic session is requested by the Tester, the ECU shall first send a positive response messsage startDiagnosticSession before the new session is activated. If the ECU sends a negative response message following a startDiagnosticSession request, the currently active session shall continue.

6.1

startDiagnosticSession service

Message is used to enable different diagnostic sessions, when no diagnostic session has been requested, the default diagnostic session is activated.

6.1.1

Message data bytes

startDiagnosticSession Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	startDiagnosticSession Request service Id	M	10
#2	DiagnosticMode=[See § 6.1.2.1]	M	xx
#3	BaudRate Ik =[See § 6.1.2.2]	U	xx

startDiagnosticSession Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	startDiagnosticSession Positive Response service Id	S	50
#2	Diagnostic Mode =[See § 6.1.2.1]	M	xx

startDiagnosticSession Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	startDiagnosticSession Request service Id	M	10
#3	ResponseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

6.1.2

Parameter Definitions

6.1.2.1

Diagnostic Mode

The DiagnosticMode parameter identifies the requested diagnostic session.

Hex Value	ResponseCode	Used in this Application
81	DefaultMode−StandardDiagnosticMode−OBDIIMode	✓
83	EOL Fiat	TBD
84	EndOfLineSystemSupplierMode	TBD
85	ECUProgrammingMode	TBD
89	Dedicated session for component starting	TBD
8A−F9	Reserved by FIAT	TBD

The ECU Supplier can not use the a.m. diagnosticModes to improve his services. For that purpose, it will be possible to use the endOfLineSystemSupplierMode (84h) or systemSupplierSpecific (FAh−FEh).

The diagnosticMode 89h must be used by the ECUs (f.e. NCL) in which the components start during default diagnosis session (81) is technically impossible or in case the result is different from Client requirement.

6.1.2.2

BaudRate Ik

The BaudRate Ik parameter identifies the baud rate to be activated after the positive response; the absence of this parameter from the request block indicates that no change is requested in the current baud rate.

Hex Value	Baud Rate (Baud)	Used in this Application
01	9600	TBD
02	19200	✓
03	38400	✓
04	57600	TBD
05	115200	TBD
80−F9	Reserved by FIAT	TBD
FA−FE	Reserved by system supplier	TBD

6.1.2.3

DiagnosticService List

The following table shows how commands are subdivided within each diagnostic session.

	DiagnosticMode	
Diagnostic service	**DefaultMode – StandardDiagnosticMode – OBDIIMode**	**ECUProgrammingMode**
accessCommunicationParameter	✓	✓
startDiagnosticSession	✓	✓
stopDiagnosticSession	✓	✓
SecurityAccess	✓	✓
testerPresent	✓	✓
ReadECUIdentification	✓	✓
ReadDataByLocalIdentifier	✓	✓
WriteDataByLocalIdentifier	✓	✓
ReadDiagnosticTroubleCodesByStatus	✓	
ReadStatusOfDiagnosticTroubleCodes	✓	
ReadFreezeFrameData	✓	
ClearDiagnosticInformation	✓	
InputOutputControlByLocalIdentifier	✓	
StartRoutinebyLocalIdentifier	✓	✓
StopRoutineByLocalIdentifier	✓	✓
RequestRoutineResultByLocalIdentifier	✓	✓
RequestDownload		✓
TransferData		✓
RequestTransferExit		✓

6.2

stopDiagnosticSession service

On receiving this command, the ECU shall activate the default diagnostic session or disable the current diagnostc mode in the ECU.

6.2.1

Message data bytes

stopDiagnosticSession Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	stopDiagnosticSession Request service Id	M	20

stopDiagnosticSession Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	stopDiagnosticSession Positive Response service Id	S	60

stopDiagnosticSession Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	stopDiagnosticSession Request service Id	M	20
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

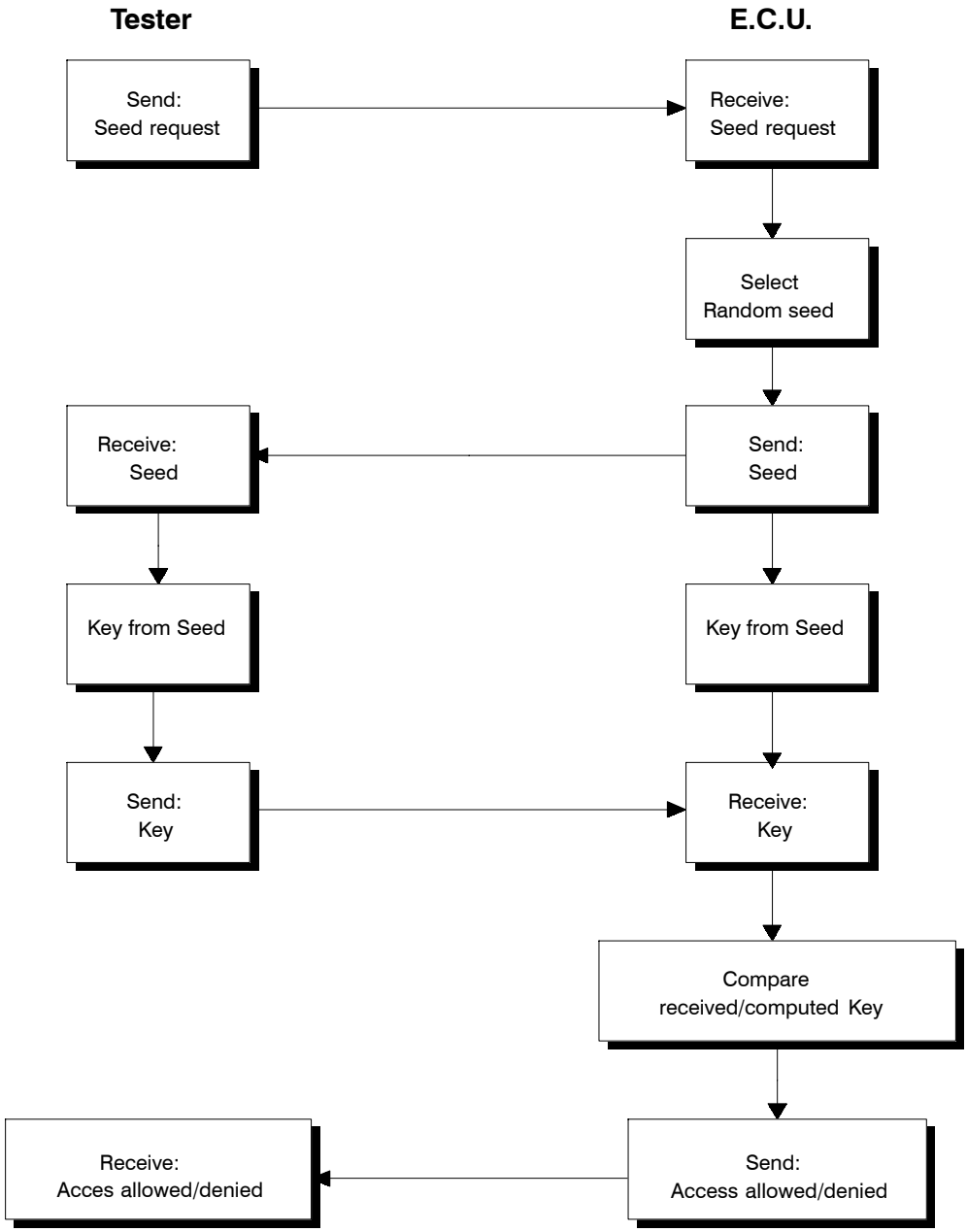
6.3

securityAccess service

This service shall be used to improve the safety measures during data exchange, in order to protect some ECU confidential or critical functionality.

Following are some general rules to be observed:

● The Tester shall request the ECU to enable the execution of security−protected commands or services ("unlock") by sending service securityAccess request #1. The ECU shall answer by sending a seed using service securityAccess positive response #1. The Tester shall continue returning a key back to the ECU using the service securityAccess request #2. The ECU shall compare this key to one internally stored. If the two keys match, the ECU shall enable the Tester for the requested operations and notify this using service securityAccess positive response #2. If upon 2 attempts the command securityAccess request #2 still fails (keys don't match), the ECU shall insert a 10 seconds time delay before allowing further attempts.



- No additional delay is accepted before the ECU responds to a securityAccess after power−on.
- If the ECU, once it has been unlocked by the securityAccess procedure, receives another SecurityAccess request #1 command, it shall respond with service securityAccess positive response #1 with seed set to ”$00 00”. The Tester may use this method in order to know the ECU status (locked/unlocked).
- The security system is not supposed to protect normal diagnostic operations but it shall simply protect restricted data areas or download procedures, in order to provide the ECU with software protection against unauthorized intrusions.
- ECU’s providing the security algorithm shall reject requests of protected services when they are locked.
- The securityAccess Mode and the security algorithm are related to the type of operation to be carried out.
- The security algorithm is not covered in this document for security reasons.

6.3.1

Message data bytes

securityAccess #1 Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	securityAccess #1 Request service Id	M	27
#2	accessMode = [requestSeed]	M	xx
#3	accessParameter	U	xx

securityAccess Positive Response #1 Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	securityAccess Positive Response #1 service Id	S	67
#2	accessMode = [requestSeed]	M	xx
#3	seed#1 (High Byte)	C	xx
:	:	:	:
#n	seed#m (Low Byte)	C	xx

C : accessMode deve essere settato come ”requestSeed”

securityAccess Negative Response #1 Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negative Response service Id	S	7F
#2	securityAccess Request service Id	M	27
#3	responseCode = [See § 5.2.2 − Parameter Definitions]	M	xx

securityAccess #2 Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	securityAccess #2 Request service Id	M	27
#2	accessMode = [sendKey]	M	xx
#3	key#1 (High Byte)	C	xx
:	:	:	:
#n	key#m (Low Byte)	C	xx

C : accessMode must be set to ”sendKey”

securityAccess Positive Response #2 Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	securityAccess Positive Response #2 service Id	S	67
#2	accessMode = [sendKey]	M	xx
#3	SecurityAccessStatus = [SecurityAccessAllowed]	M	34

securityAccess Negative Response #2 Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negative Response service Id	S	7F
#2	securityAccess Request service Id	M	27
#3	ResponseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

6.3.2

Parameter Definitions

6.3.2.1

AccessMode

The AccessMode parameter identifies the requested security level.

Hex Value	AccessMode
01,03,05–7F	Seed request with multiple security levels defined by FIAT
02,04,06–7E	Key send with multiple security levels defined by FIAT

6.4

testerPresent service

This service is used to indicate to the ECU that the Tester is present in order to avoid that the ECU automatically resumes its normal operation mode and terminates communication.

6.4.1

Message data bytes

testerPresent Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	testerPresent Request service Id	M	3E

testerPresent Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	testerPresent Positive Response service Id	S	7E

testerPresent Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	testerPresent Request service Id	M	3E
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

6.5

ecuReset service

The service ecuReset is not used in the FIAT implementation of protocol KWP2000.

6.6

readEcuIdentification service

This service is used to read the ECU identification data.

6.6.1

Message data bytes

readEcuIdentification Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readEcuIdentification Request service Id	M	1A
#2	IdentificationOption = [See § 6.6.2.1 – Identification Option]	M	xx

readEcuIdentification Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readEcuIdentification Positive Response service Id	S	5A
#2	identificationOption = [See § 6.6.2.1 – Identification Option]	M	xx
#3	ECUIdentificationParameter#1	C	xx
:	:	:	:
#n	ECUIdentificationParameter#m	C	xx

C : parameters depend on the identificationOption value.

readEcuIdentification Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negative Response service Id	S	7F
#2	readEcuIdentification Request service Id	M	1A
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

6.6.2

Parameter Definitions

6.6.2.1

IdentificationOption

The IdentificationOption parameter identifies which parameter of the identification code is requested.

Hex Value	Identification Option	ECU Identification Data Format	# of bytes	Data Format	Used in this application
80	Identification Code	All data 91..99	61	– – –	✓
91	FIAT drawing no.	'123456789'	11	ASCII	✓
92	ECU hardware no.	TBD	11	ASCII	✓
93	ECU hardware version no.	TBD	1	UNSGN	✓
94	ECU software no.	TBD	11	ASCII	✓
95	ECU software version no.	TBD	2	UNSGN	✓
96	Omologation code	TBD	6	ASCII	✓
97	ISO code	xx xx xx xx xx [See § 6.6.2.2]	5	UNSGN	✓
98	Tester code	TBD	10	ASCII	✓
99	reprogramming/production date	19 97 03 31 [Y–Y–M–D]	4	BCD	✓

Parameters identified by codes 91h, 92h, 94h, 96h and 98h shall be aligned to the left: bytes which are not used in these fields shall be filled with blanks (20h). If options 93h and 95h are not used, the corresponding fields shall be filled with zeroes (00h).

Parameter 98h shall be filled with blanks (20h) by Supplier when the ECU is shipped to FIAT; it shall be possible for the Tester to re–write this field each time the ECU is completely re–programmed (downloaded). In those ECU's in which the download function is not implemented, said field shall nevertheless be filled by Supplier at production time.

Parameter 99h shall be supplied with the ECU production date when the ECU is shipped to FIAT. It shall be possible for the Tester to re–write this field each time the ECU is completely re–programmed (downloaded). In those ECU's in which the download function is not implemented, said field shall nevertheless filled by Supplier at production time.

6.6.2.2

Definition of the ISO code

It is the responsibility of FIAT to assign a unique ISO code identifying the diagnostic performance of the ECU. The ISO shall be correctly written into the ECU under the complete responsibility of the Supplier. The key bytes #1/#2 are assigned according to ISO Std. 9141, while bytes #4 and #5 shall conform to FIAT DT SIEE SSE.

The ECU Supplier shall ask Fiat for a new ISO code whenever a change relevant to any of the following points produces a change in the diagnostic functionality:

- New ECU or hardware modifications
- Changes in the pinout
- Changes in the diagnostic protocol
- New sensors
- Addition/removal of sensors
- Changes in conversion formula(e).
- New or modified information exchanged through the diagnostic protocol.

Byte #1	Byte #2	Byte #3	Byte #4	Byte #5	Vehicle
xx	xx	xx	xx	xx	TBD

7

DATA TRANSMISSION FUNCTIONAL UNIT

7.0

Introduction

Services described in this functional unit shall be used by the Tester to read/write data in the ECU memory. Access shall take place through local identifiers.

7.1

reaDataByLocalIdentifier service

This service shall be used by the Tester to access some internal parameters of the ECU reading their current value. Access to restricted parameters shall follow a successful security access sequence.

7.1.1

Message data bytes

readDataByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readDataByLocalIdentifier Request service Id	M	21
#2	recordLocalIdentifier = [See § 7.1.2.1 – RLI List]	M	xx

readDataByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readDataByLocalIdentifier Positive Response service Id	S	61
#2	recordLocalIdentifier = [See § 7.1.2.1 – RLI List]	M	xx
#3	RecordValue #1	M	xx
	:	:	:
#n	recordValue #m	M	xx

readDataByLocalIdentifier Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	readDataByLocalIdentifier Request service Id	M	21
#3	responseCode = [See § 5.2.2 Parameter Definitions]	M	xx

responseCode = 33h in the readDataByLocalIdentifier negative response message shall be used when the requested recordLocalIdentifier is protected by security access and the Tester has not yet requested the ECU to unlock access (through a seed–and–key procedure).

7.1.2

Parameter Definitions

7.1.2.1

RLI List

The following table defines the general criteria used to assign values to recordLocalIdentifier

RLI (Hex)	Description	Read	Write	Num. of Bytes	Conversion
01−1F	Reserved by this document				See 07234
20−2F	EEPROM data	✓	TBD	2	TBD
30−7F	RAM data	✓	TBD	2	TBD
980	Tester Code	NO	✓	See § 6.6.2.1 Identification Option	See § 6.6.2.1 Identification Option
99	Production or Programming date	NO	✓	See § 6.6.2.1 Identification Option	See § 6.6.2.1 Identification Option
9A−9F	Reserved for identificationOption	NO	NO		
A0−BF	snapshots	✓	NO	TBD	TBD
C0−FE	not used				

All information shall be accessible through the readDataByLocalIdentifier command and shall have length of 2 bytes preferably.
Variables indicating a status (e.g. digital inputs/outputs, system status, etc.) shall be treated as 1−bit fields and grouped into a status byte/word.

All the available information shall be accessible as individual parameters, provided the ECU has enough resources and the data can be inserted into snapshots, maintaining the same data length and conversion formula.

7.2

readDataByCommonIdentifier service

The service readDataByCommonIdentifier is not used in the FIAT implementation of protocol KWP2000.

7.3

readMemoryByAddress service

The service readMemoryByAddress is not used in the FIAT implementation of protocol KWP2000.

7.4

dynamicallyDefinedLocalIdentifier service

The service dynamicallyDefinedLocalIdentifier is not used in the FIAT implementation of protocol KWP2000.

7.5

writeDataByLocalIdentifier service

This service is to be used by the Tester in order to access some ECU parameters and update current values. The access to restricted parameters shall be conditioned by a successful security access sequence.

7.5.1

Message data bytes

writeDataByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	writeDataByLocalIdentifier Request service Id	M	3B
#2	recordLocalIdentifier = [See § 7.1.2.1 – RLI List]	M	xx
#3 : #n	recordValue#1 : recordValue#m	M : M	xx : xx

writeDataByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	writeDataByLocalIdentifier Positive Response service Id	S	7B
#2	recordLocalIdentifier = [See § 7.1.2.1 – RLI List]	M	xx

writeDataByLocalIdentifier Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	writeDataByLocalIdentifier Request service Id	M	3B
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

7.6

writeDataByCommonIdentifier service

The service writeDataByCommonIdentifier is not used in the FIAT implementation of protocol KWP2000.

7.7

writeMemoryByAddress service

The service writeMemoryByAddress is not used in the FIAT implementation of protocol KWP2000.

8

STORED DATA TRANSMISSION FUNCTIONAL UNIT

8.0

Introduction

The services provided by this functional unit are to be used by the Tester in order to read the current content of the ECU error memory. The ECU shall make sure that data sent to the Tester are updated to the moment of the request. The ECU must also take appropriate measures against the possibility of losing the error memory contents in case of key−off and power−off.

8.1

readDiagnosticTroubleCodes service

The service readDiagnosticTroubleCodes is not used in the FIAT implementation of protocol KWP2000.

8.2

readDiagnosticTroubleCodesByStatus service

This service is to be used by the Tester to read the complete list of diagnosticTroubleCodes stored in the error memory of the ECU at the moment of the request, independently of their current status.

8.2.1

Message data bytes

readDiagnosticTroubleCodesByStatus Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	ReadDiagnosticTroubleCodesByStatus Request service Id	M	18
#2	StatusOfDTC = requestIdentifiedDTCAndStatus	M	00
#3	groupOfDTC (High Byte)	M	FF
#4	groupOfDTC (Low Byte) = [all groups]	M	00

readDiagnosticTroubleCodesByStatus Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readDiagnosticTroubleCodesByStatus Pos.Resp. Serv. Id	M	58
#2	numberOfDTC	M	xx
#3	listOfDTCAndStatus = [	C	
	DTC #1 (High Byte)		xx
	DTC #1 (Low Byte)		xx
	statusOfDTC #1		xx
	DTC #m (High Byte)		xx
	DTC #m (Low Byte)		xx
#n	statusOfDTC #m		xx
	]		

C : listOfDTCAndStatus is present only if OfDTC > 0

readDiagnosticTroubleCodesByStatus Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	readDiagnosticTroubleCodesByStatus Request service Id	M	18
#3	responseCode = [See § 5.2.2 − Parameter Definitions]	M	xx

If the ECU has no DTC with stored status information, it must send a positive response message with numberOfDTC set to 0h, without including the parameters of listOfDTCAndStatus.

8.2.2

Parameter Definitions

8.2.2.1

groupOfDTC

The parameter groupOfDTC indicates which error group is requested.

High Byte (Hex)	Low Byte (Hex)	Description
FF	00	All groups

8.2.2.2

numberOfDTC

The parameter numberOfDTC indicates the number of errors present in memory at the moment of request and could range from 0 up to the max. number of stored errors.

A value of 0 indicates that there are no errors stored.

8.2.2.3

Error Memory structure

The error memory is to be divided in 10−byte blocks, each containing the information required to describe the error code stored by the system, in the following format:

Block	Byte	Description	Conversion	Units
#1	#1 #2	DTC (High Byte) DTC (Low Byte)	See § 8.2.2.1 groupOfDTC	See § 8.2.2.1 groupOfDTC
	#3	statusOfDTC	See § 8.2.2.5	See § 8.2.2.5
	#4	environmentalCondition #1	TBD	TBD
	#5	environmentalCondition #2	TBD	TBD
	#6	environmentalCondition #3	TBD	TBD
	#7	environmentalCondition #4	TBD	TBD
	#8	environmentalCondition #5	TBD	TBD
	#9	environmentalCondition #6	TBD	TBD
	#10	eventCounter	1 bit = 1 event	[events]
#2	#1 #2	DTC (High Byte) DTC (Low Byte)	See § 8.2.2.1 groupOfDTC	See § 8.2.2.1 groupOfDTC
	:	:	:	:
	#10	eventCounter	1 bit = 1 event	[events]
...	:	:	:	:
#n	#1 #2	DTC (High Byte) DTC (Low Byte)	See § 8.2.2.1 groupOfDTC	See § 8.2.2.1 groupOfDTC
.	:	:	:	:
	#10	eventCounter	1 bit = 1 event	[events]

The min. number of stored blocks shall be equal or higher than 5; max. number of stored blocks is to be agreed upon by FIAT and the ECU supplier.

If all cells are occupied and one further error is detected, the non−current error with the lowest event-Counter shall be replaced; if two or more non−current errors have the same eventCounter the one with the lowest priority shall be replaced.

The list of priorities shall be provided by the ECU Supplier.

Whenever the ECU detects an error whose DTC is already present in memory, the ECU beahviour shall be the following:

- the fields of DTCStorageData (bit 6 and 5) and DTCReadinessFlag (bit 4) in statusOfDTC must be updated, while all other information must not be changed.
- environmental conditions are not to be updated.
- for meter management see Standard FIAT 07234 Chapt. 4.3
  for EOBD systems see Chapt. 4.3.7

The **DTC Code** identifies the faulty component [See § 8.2.2.4 − Tables for DTC and environmental-Conditions]

statusOfDTC identifies the fault status [See § 8.2.2.5 − StatusOfDTC]

environmentalCondition bytes are parameters which are stored in the ECU at the moment a fault is detected for the first time, and shall be always the same independently of error type.

eventCounter shall be initialized to value 64 when the error is first detected and shall be decreased by 1 unit avery time a complete cycle is executed without the occurrence of any anomaly **(*)**.
When the eventCounter reaches the value of 0 the information related to the fault shall be erased.
The conditions for the confirmation/cancellation of a fault shall be agreed upon between FIAT and the ECU Supplier.

(*) The definition of “cycle” shall be agreed upon between FIAT and the ECU Supplier.

8.2.2.4

Tables for DTC and environmentalConditions

The following rules and tables shall be used for the definition of DTC’s and of the associated environmentalCondition parameters.

DTC and DTC groups

DTC’s are divided in 5 groups: POWERTRAIN, CHASSIS, BODY, NETWORK COMMUNICATION, ALL (ALL = all vehicle systems) according to SAE J 2012 − Diagnostic Trouble Code.

DTC’s are of the BCD type (binary coded decimal).

The following table specifies the possible values for this parameter.

High Byte (Hex)	Low Byte (Hex)	Description
FF	00	All groups (A)
00	00	Powertrain group (P) − DTC Powertrain from 0001h to 3999h
40	00	Chassis group (C) − DTC Chassis from 4001h to 7999h
80	00	Body group (B) − DTC Body from 8001h to B999h
C0	00	Network group (U) − DTC Network from C001h to F999h

Rules for the assignment of the diagnosticTroubleCode

The DTC parameter is used by the ECU to bring to attention system faults by means of a two−byte BCD number. The format of a DTC is specified in /6/ . Decoding is shown in table below.

Bits 15,14: P,C,B,U/A	Bits 13,12: sub−groups 0−3	DTC Number (Binary Coded Decimal) fields: 001 − 999		
0 0 (**P**)owertrain	0	0	0	1
:	:	:	:	:
0 0 (**P**)owertrain	3	9	9	9
0 1 (**C**)hassis	0	0	0	1
:	:	:	:	:
0 1 (**C**)hassis	3	9	9	9
1 0 (**B**)ody	0	0	0	1
:	:	:	:	:
1 0 (**B**)ody	3	9	9	9
1 1 (**U**)Network/All	0	0	0	1
:	:	:	:	:
1 1 (**U**)Network/All	3	9	9	9

Sub−group 0 shall be used by the DTC's controlled by ISO/SAE for which it has been possible to obtain a uniform definition.
Sub−group 3 is reserved: DTC's in this group shall not be used by the ECU.

FIAT encoding of DTC's follows the rules listed below:

• DTC's shall belong to sub−groups 1 or 2 ("manufacturer controlled") unless otherwise specified.

• A DTC identifies an ECU sub−component which is to be replaced as one piece during service. Ex.: it shall not be necessary to distinguish beteween an ECU with a faulty memory bank and one having I/O trouble, since in both cases service personnel shall replace the whole ECU. In this case, the same DTC shall be used with different DTCFaultSymptoms, in order to reduce the total number of DTC's.

The following table shall be used as a reference in order to specify all DTC's defined for one specific ECU.

DTC Code	Description	DTC Fault Symptom
TBD	TBD	TBD

Table below shall be used as a reference in order to specify the environmentalConditions parameters defined for a specific ECU.

Position	Description	Conversion formula
1	TBD	TBD
2	TBD	TBD
3	TBD	TBD
4	TBD	TBD
5	TBD	TBD
6	TBD	TBD

8.2.2.5

StatusOfDTC

The following table contains the description of the field statusOfDTC, which shall be strictly observed. Each variations shall be agreed upon between FIAT and the ECU supplier.

SODTC−RE **Bit: 7654 3210b**	**Description of StatusOfDTC−Response**
WBAT **SSSS**b	**DTCFaultSymptom {'S' bit : 3 − 0 } (*)**
{ $x0 } 0000	**no fault symptom available for this DTC**
{ $x0 } 0001	**above maximum threshold**
{ $x0 } 0010	**below minimum threshold**
{ $x0 } 0100	**no signal**
{ $x0 } 1000	**invalid signal**
WBA**T** SSSSb	**DTCReadinessFlag {'T' bit : 4 }**
0	**test complete for this DTC or not applicable**
1	**test not complete for this DTC**
W**BA**T SSSSb	**DTCStorageState { B − bit 6, A = bit 5}**
0 0	**noDTCDetected at time of request** no DTC stored in non−volatile memory
0 1	**DTCNotPresent at time of request** A DTC was present. DTC stored in non−volatile memory.
1 0	**DTCMaturing−Intermittent at time of request** Insufficient data to consider the error ready for storage in non−volatile memory.
1 1	**DTCPresent at time of request** DTC stored in non−volatile memory
WBAT SSSSb	**DTCWarningLampCalibrationStatus {'W' bit: 7 }**
0	**disabled (Warning Lamp is not illuminated for this DTC)**
1	**enabled (Warning Lamp illuminated for this DTC)**

(*) Codes x3h,x5h,x6h,x7h,x9h,xAh,xBh,xCh,xDh,xEh,xFh may be used following a specific agreement between FIAT and Supplier.

8.3

readStatusOfDiagnosticTroubleCodes service

This service shall be used by the Tester to read details relative to a diagnosticTroubleCode stored in the error memory of the ECU at any given moment.
Tester is not authorized to use this service to read two or more data sets relative to a diagnosticTroubleCode in the same message.

8.3.1

Message data bytes

readStatusOfDiagnosticTroubleCodes Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readStatusOfDiagnosticTroubleCodes Request service Id	M	17
#2	DTC (High Byte)	M	xx
#3	DTC (Low Byte) = [See §8.2.2.4 – Tables for DTC and environmental Conditions]	M	xx

readStatusOfDiagnosticTroubleCodes Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readStatusOfDiagnosticTroubleCodes Positive Response service Id	S	57
#2	numberOfDTC	M	xx
	listOfDTCAndStatus = [	C	
#3	DTC (High Byte)		xx
#4	DTC (Low Byte)		xx
#5	statusOfDTC		xx
#6	environmentalCondition#1		xx
#7	environmentalCondition#2		xx
#8	environmentalCondition#3		xx
#9	environmentalCondition#4		xx
#10	environmentalCondition#5		xx
#11	environmentalCondition#6		xx
#12	eventCounter#7		xx
	]		

C : The parameters of listOfDTCAndStatus are present only if numberOfDTC > 0

readStatusOfDiagnosticTroubleCodes Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	readStatusOfDiagnosticTroubleCodes Request service Id	M	17
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

If the requested DTC is not stored in the memory of the ECU when the request is received, the ECU shall send a Positive Response Message with a numberOfDTC = 00h, excluding also the parameter listOfDTCAndStatus.

The length of the readStatusOfDiagnosticTroubleCodes positive response with numberOfDTC=1 is fixed to 12 bytes: if the ECU does not support 6 environmentalCondition parameters for its DTC's, it shall fill the listOfDTCAndStatus with zeroes (00h).

The ECU shall use the negative response with responseCode set to 12h when the Tester requests an unknown DTC.

NOTE: The numberOfDTC parameter may take up the values of 0 or 1; a value of 0 indicates that the requested DTC is recognized but not present. Value 1 indicates that the requested DTC is present and available.

8.4

readFreezeFrameData service

This service shall be used to access the “Freeze Frame” data on ECU’s which must submit to EOBD specifications.

8.4.1

Message data bytes

readFreezeFrameData Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readFreezeFrameData Request service Id	M	12
#2	freezeFrameNumber	M	01
#3	recordAccessMethodIdentifier = requestAllData	M	00

readFreezeFrameData Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	readFreezeFrameData Positive Response service Id	S	52
#2	freezeFrameNumber	M	01
#3	freezeFrameData#1	M	xx
:	:	:	:
#n−1	freezeFrameData#m	M	xx
#n	recordAccessMethodIdentifier = requestAllData	M	00

readFreezeFrameData Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	readFreezeFrameData Request service Id	M	12
#3	responseCode = [See § 5.2.2 − Parameter Definitions]	M	xx

8.5

ClearDiagnosticInformation service

The messages described below shall be used by the Tester to clear all diagnostic information about any diagnosticTroubleCode currently stored in the ECU's memory.
The Tester is not allowed to use this service to partially reset the contents of the error memory.

8.5.1

Message data bytes

clearDiagnosticInformation Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	clearDiagnosticInformation Request service Id	M	14
#2	groupOfDiagnosticInformation (High Byte)	M	FF
#3	groupOfDiagnosticInformation (Low Byte) = [all groups]	M	00

clearDiagnosticInformation Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	clearDiagnosticInformation Positive Response service Id	S	54
#2	groupOfDiagnosticInformation (High Byte)	M	FF
#3	groupOfDiagnosticInformation (Low Byte) = [all groups]	M	00

clearDiagnosticInformation Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	clearDiagnosticInformation Request service Id	M	14
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

9

INPUT/OUTPUT CONTROL FUNCTIONAL UNIT

9.0

Introduction

This functional unit contains services which can be used by the Tester to control specific inputs/outputs of an ECU.

9.1

inputOutputControlByLocalIdentifier service

This service shall be used to replace an input signal value, an internal function of the ECU and/or control an output (actuator) of an ECU by referring to a local identifier.

9.1.1

Message data bytes

inputOutputControlByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Request service Id	M	30
#2	inputOutputLocalIdentifier = [See § 9.1.2.3]	M	xx
#3	controlStatus = [inputOutputControlParameter = [See § 9.1.2.1]	M	xx
#4	controlState#1 [See § 9.1.2.2]	C	xx
:		:	:
#n	controlState#m [See § 9.1.2.2]]	C	xx

C : controlState only if inputOutputControlParameter=07h or 08h

inputOutputControlByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Pos.Resp. Serv. Id	S	70
#2	inputOutputLocalIdentifier = [See § 9.1.2.3]	M	xx
#3	controlStatus = [inputOutputControlParameter = [See § 9.1.2.1]	M	xx
#4	controlState#1 [See § 9.1.2.2]	C	xx
:		:	:
#n	controlState#m [See § 9.1.2.2]]	C	xx

C : controlState is present only if inputOutputControlParameter=01h.

inputOutputControlByLocalIdentifier Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	inputOutputControlByLocalIdentifier Request service Id	M	30
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

9.1.2

Parameter Definitions

9.1.2.1

inputOutputControlParameter

This service shall be used to replace an input signal value, an internal function of the ECU and/or control an output (actuator) of an ECU by referring to a local identifier.

Hex Value	Parameter Name and Description
00	**returnControlToECU** This parameter indicates that the ECU has to regain complete control of the input signal, internal parameter or output signal identified by InputOutputLocalIdentifier.
01	**reportCurrentState** This parameter indicates that the ECU has to report the state of the input signal, internal parameter or output signal identified by InputOutputLocalIdentifier.
04	**resetToDefault** This parameter indicates that the ECU is requested to reset the input signal, internal parameter or output signal identified by InputOutputLocalIdentifier to the default value.
07	**shortTermAdjustment** This value indicates that the ECU is requested to adjust the input signal, internal parameter or output signal referenced by the InputOutputLocalIdentifier in RAM to the value specified in ControlOptionParameter (e.g. set idle actuator to a specified step number, set a PWM signal to a specified value).
08	**longTermAdjustment** This value indicates that the ECU is requested to adjust the input signal, internal parameter or output signal referenced by the InputOutputLocalIdentifier in EEPROM/FLASH EEPROM to the value specified in ControlOptionParameter (e.g. set engine idle speed, set CO).

9.1.2.2

inputOutputControlState

The parameter inputOutputControlState indicates the state the component has to be brought to. The values available in the FIAT implementation of KWP2000 are listed below.

If this parameter is not inserted in the request service, the ECU shall actuate the automatic test mode.

Hex Value	Description
00	Component is set to OFF state (ON/OFF components)
FF	Component is set to ON state (ON/OFF components)
00 (000%) FF (100%)	Component is set to a definite state (%) (Components with control capability)

In case of components which allow only a default activation (e.g.: one ON/OFF cycle with specified duty and timing) it is necessary to use inputOutputControlParameter = 07h (shortTermAdjustment) without controlState parameter.

Example of activation to ON of a component :

inputOutputControlByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Request service Id	M	30
#2	inputOutputLocalIdentifier	M	xx
#3	inputOutputControlParameter	M	07
#4	controlState#1	C	FF

inputOutputControlByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier PosResp. Serv. Id	S	70
#2	inputOutputLocalIdentifier	M	xx
#3	inputOutputControlParameter	M	07

Example of activation to OFF of a component :

inputOutputControlByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Request service Id	M	30
#2	inputOutputLocalIdentifier	M	xx
#3	inputOutputControlParameter	M	07h
#4	controlState#1	C	00h

inputOutputControlByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Pos.Resp. Serv. Id	S	70
#2	inputOutputLocalIdentifier	M	xx
#3	inputOutputControlParameter	M	07

Example of activation of a component in automatic mode:

inputOutputControlByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Request service Id	M	30
#2	inputOutputLocalIdentifier	M	xx
#3	inputOutputControlParameter	M	07

inputOutputControlByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Pos.Resp. Serv. Id	S	70
#2	InputOutputLocalIdentifier	M	xx
#3	inputOutputControlParameter	M	07

Example of returning control to the ECU:

inputOutputControlByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Request service Id	M	30
#2	inputOutputLocalIdentifier	M	xx
#3	inputOutputControlParameter	M	00

inputOutputControlByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	inputOutputControlByLocalIdentifier Pos.Resp. Serv. Id	S	70
#2	InputOutputLocalIdentifier	M	xx
#3	inputOutputControlParameter	M	00

9.1.2.3

inputOutputLocalIdentifier

The table below shall be used to list all the inputOutputLocalIdentifier values and, for each component, the possible InputOutputControlState.

InputOutputLocalIdentifier		Applicable inputOutputControlState Values				
Hex Value	**Component Description**	**00h**	**01h**	**04h**	**07h**	**08h**
TBD	TBD	TBD	TBD	TBD	TBD	TBD

9.1.2.4

IOLI Details

Hereafter are described some general criteria :

The environmental conditions enabling activation shall be agreed upon between FIAT and ECU Supplier; if such conditions are not observed, the ECU shall immediately regain control.

The activation of a component by means of inputOutputControlByLocalIdentifier service shall be interrupted when any of the following conditions comes true:

- The ECU has not received any other inputOutputControlByLocalIdentifier command since > 30s.
- The ECU receives a stopDiagnosticSession command.
- The ECU receives a stopCommunication command.
- The ECU receives a time−out during the diagnostic session.

If any of the conditions mentioned above is met the ECU shall regain immediate control of the component.

The ECU Supplier shall agree with FIAT any additional restrictive conditions.

9.2

inputOutputControlByCommonIdentifier service

The service inputOutputControlByCommonIdentifier is not used in the FIAT implementation of protocol KWP2000.

10

REMOTE ACTIVATION OF ROUTINE FUNCTIONAL UNIT

10.0

Introduction

This functional unit specifies the remote activation services of routines and the way they are to be implemented by the ECU's and by the Tester. The possible implementation methods are many. The method adopted by FIAT for the KWP2000 on K−line is based on the assumption that after a routine has started in the ECU memory, following a request by the Tester, the ECU itself is responsible for stopping its execution.

- The ECU routine shall start within a time comprised between the end of the request message startRoutineByLocalIdentifier and the end of the first response message.
- The Tester can request the interruption of a routine using request message stopRoutineByLocalIdentifier
- The Tester shall use service requestroutineResulByLocalIdentifier to wait for the end of the routine and obtain its exit information.
- During the execution of the routine the ECU shall use negative response requestRoutineResultByLocalIdentifier with response codes 23h (routineNotComplete) and 21h (busy−repeatRequest) to indicate to the Tester that the routine is under way but not yet completed (See § 5.2.2).

10.1

startRoutineByLocalIdentifier service

This service shall be used by the Tester to start execution of a routine in the ECU memory. The routine is indicated by a Local Identifier.

10.1.1

Message data bytes

startRoutineByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	startRoutineByLocalIdentifier Request service Id	M	31
#2	routineLocalIdentifier = [See § 10.1.2]	M	xx
#3	routineEntryOption#1	U	xx
#n	routineEntryOption#m	U	xx

startRoutineByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	startRoutineByLocalIdentifier Pos.Resp. Serv. Id	S	71
#2	routineLocalIdentifier = [See § 10.1.2]	M	xx

startRoutineByLocalIdentifier Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	StartRoutineByLocalIdentifier Request service Id	M	31
#3	responseCode = [See § 5.2.2 − Parameter Definitions]	M	xx

10.1.2

Parameter Definitions

The table below shows the values currently defined for parameter routineLocalIdentifier in the FIAT implementation of KWP2000 on K−line, offering a description of the associated routines and related routineEntryOptions parameters.
The routines indicated are used during download procedures. ECU's not implementing download functionality do not have to implement these routines, and shall avoid use of their routineLocalIdentifiers.

RELI (Hex)	Routine description	routineEntryOptions #	routineEntryOptions description
01	Checksum calculation	1	start address (high byte)
		2	start address (middle byte)
		3	start address (low byte)
		4	stop address (high byte)
		5	stop address (middle byte)
		6	stop address (low byte)
		7	expected checksum (high byte)
		8	expected checksum (low byte)
02	FLASH memory erasure	1	start address (high byte)
		2	start address (middle byte)
		3	start address (low byte)
		4	stop address (high byte)
		5	stop address (middle byte)
		6	stop address (low byte)
TBD	TBD	TBD	TBD

10.2

startRoutineByAddress service

The service startRoutineByAddress is not used in the FIAT implementation of protocol KWP2000.

10.3

stopRoutineByLocalIdentifier service

This service has to be used by the Tester in order to stop the execution of a routine in the memory of the ECU. The routine is indicated by means of a Local Identifier.

10.3.1

Message data bytes

stopRoutineByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	stopRoutineByLocalIdentifier Request service Id	M	32
#2	routineLocalIdentifier = [See § 10.1.2]	M	xx

stopRoutineByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	startRoutineByLocalIdentifier Pos.Resp. Serv. Id	S	72
#2	routineLocalIdentifier = [See § 10.1.2]	M	xx

stopRoutineByLocalIdentifier Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	stopRoutineByLocalIdentifier Request service Id	M	32
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

10.4

stopRoutineByAddress service

The service stopRoutineByAddress is not used in the FIAT implementation of protocol KWP2000.

10.5

requestRoutineResultsByLocalIdentifier service

This service shall be used by the Tester to request the results of a routine.
The ECU shall use a negative response message with response codes 23h (routineNotComplete) and 21h (busy−repeatRequest) to indicate that the routine is under way but has not yet been completed (See § 5.2.2).

10.5.1

Message data bytes

requestRoutineResultsoutineByLocalIdentifier Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	requestRoutineResultsByLocalIdentifier Request service Id	M	33
#2	routineLocalIdentifier = [See § 10.1.2]	M	xx

requestRoutineResultsByLocalIdentifier Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	requestRoutineResultsByLocalIdentifier Positive Response service Id	S	73
#2	routineLocalIdentifier = [See § 10.1.2]	M	xx
#3	routineResults#1	M	xx
:	:	:	:
#n	routineResults#m	U	xx

requestRoutineResultsByLocalIdentifier Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	requestRoutineResultsByLocalIdentifier Request service Id	M	33
#3	responseCode = [See § 5.2.2 − Parameter Definitions]	M	xx

10.5.2

Parameter Definitions

The table below shall list the routines supported by the ECU and related entryOption, rules and parameters returned in the positive response message.

Each routine shall be inserted in a separate paragraph with the following shown below:

Description	RELI (Hex)
TBD	TBD

List of parameters to be provided as entryOption to routine TBD:

Description TBD routineEntryOption

Data Byte #	Parameter Name	Cvt	Hex Value
#1	TBD	T	TBD
:	:	B	:
#n	TBD	D	TBD

List of parameters to be returned as routineResults by routine TBD:

Description TBD routineResults

Data Byte #	Parameter Name	Cvt	Hex Value
#1	TBD	T	TBD
:	:	B	:
#n	TBD	D	TBD

10.6

requestRoutineResultsByLocalAddress service

Service requestRoutineResultByLocalAddress is not used in the FIAT implementation of protocol KWP2000.

11

UPLOAD DOWNLOAD FUNCTIONAL UNIT

11.0

Introduction

This functional unit specifies negotiation services for data transfer as they have to be implemented by the ECU and by the Tester.

11.1

requestDownload service

This service shall be used by the Tester to initialize data transfer from the Tester to the ECU (download). After the ECU has received the request message requestDownload, it shall intiate all necessary actions in order to receive the data before sending the positive response message.

11.1.1

Message data bytes

requestDownload Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	requestDownload Request service Id	M	34
	transferRequestParameter = [		
#2	memoryAddress (High Byte)	M	xx
#3	memoryAddress (Middle Byte)	M	xx
#4	memoryAddress (Low Byte)	M	xx
#5	dataFormatIdentifier	M	xx
#6	unCompressedMemorySize (High Byte)	M	xx
#7	unCompressedMemorySize (Middle Byte)	M	xx
#8	unCompressedMemorySize (Low Byte)	M	xx
	]		

requestDownload Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	requestDownload Pos.Resp. Serv. Id	M	74
#2	transferResponseParameter = [maxNumberOfBlockLength]	M	xx

requestDownload Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	requestDownload Request service Id	M	34
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

11.1.2

Parameter Definitions

Formats supported for dataFormatIdentifier:

Hex Value	Description	Cvt
0x	unCompressed	M
1x	Bosch Compression Method	U
2x	Hitachi Compression Method	U
3x	Marelli Compression Method	U
4x	Lucas Compression Method	U
5x−Fx	TBD	U
x0	unEncrypted	M
x1	Bosch Encrypting Method	U
x2	Hitachi Encrypting Method	U
x3	Marelli Encrypting Method	U
x4	Lucas Encrypting Method	U
x5−xF	TBD	U

11.2

requestUpload service

The service requestUpload is not used in the FIAT implementation of protocol KWP2000.

11.3

transferData service

This service shall be used by the Tester in order to transfer data to the ECU.
The FIAT KWP2000 on K−line does not implement data transfer in the opposite direction:
the positive response message transferData shall not contain any parameter.

11.3.1

Message data bytes

transferData Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	transferData Request service Id	M	36
#2	TransferRequestParameter#1	U	xx
:	:	:	:
#n	transferRequestParameter#m	U	xx

transferData Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	transferData Pos.Resp. Serv. Id	M	76

transferData Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	transferData Request service Id	M	36
#3	responseCode = [See § 5.2.2 − Parameter Definitions]	M	xx

11.4

requestTransferExit service

11.4.1

Message data bytes

requestTransferExit Request Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	requestTransferExit Request service Id	M	37

requestTransferExit Positive Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	requestTransferExit Pos.Resp. Serv. Id	M	77

requestTransferExit Negative Response Message

Data byte #	Parameter Name	Cvt	Hex Value
#1	negativeResponse service Id	S	7F
#2	requestTransferExit Request service Id	M	37
#3	responseCode = [See § 5.2.2 – Parameter Definitions]	M	xx

12

SCAN TOOL

This session describes the characteristics required by FIAT for ScanTool protocol improvement, based on KWP 2000 (14230) on its own products; in particular this chapter details the differences compared to Standard 15031 and gives other clarifications.
The FIAT applications structure has max two "EOBD relevant" control units (*), connected through K line to ScanTool, engine control unit (CCM) and transmission control unit (TCU/CAE); the (*) "Freeze-Frame(FF)" is only on CCM; in case of "EOBD relevant" errors found by TCU/CAE it will start the "Mil Request" line and the CCM control unit will take on the FF data storage.

The chapters refer to Standard ISO 15031−5; f.e. "12.1 4 TECHNICAL REQUIREMENTS" refer to Standard ISO 15031−5 chapter "4. TECHNICAL REQUIREMENTS"

12.1

4. TECHNICAL REQUIREMENTS

12.1.1

4.1.2. Application timing parameter definition

12.1.1.1

4.1.2.2. Definition for ISO 14230−4

Parameter	Minimum value ms.	Maximum value ms.	Description
P2 k−line CCM	25	35	Start Response CCM frame
P2 k−line TCU/CAE	36	50	Start Response TCU/CAE frame

12.1.2

4.1.4. Data not available

12.1.2.1

4.1.4.1. ISO 14230−4 Data not available

In case of data not available or not supported, the CCM will give the correct NegativeResponse; the TCU/CAE shall not consider the input in case of data not available or not supported.

12.1.3

4.1.4.3.2. ISO 14260−4 Data not available within P2 timing

If the data are not available within P2 Max timing, the ECUs shall respond with code 0x21 (busy−RepeatRequest) instead of code 0x78 (requestCorrectlyReceived−RespondeP ending).

12.2

5. Diagnostic service definition

12.3

5.1. Service $01 – Request current powertrain diagnostic data

All parameters defined in ANNEX B of Standard ISO 15031–5, if any, shall be available via CCM.

12.4

5.2. Service $02 – Request powertrain freeze frame data

All parameters defined in ANNEX B of Standard ISO 15031–5, if any, shall be available via CCM; according to improvement FIAT TCM/CAE, they do not store any FF, then they must not consider this service.

12.5

5.3. Service $03 – Request emission–relared diagnostic information

This service is available and totally compatible with Standard 15031–5.

12.6

5.4. Service $04 – Clear/reset emission related powertrain diagnostic information

This service is available and totally compatible with Standard 15031–5.

12.7

5.5. Service $05 – Request oxygen sensor monitoring test result

This service is available and totally compatible with Standard 15031–5 only for CCM.

12.8

5.6. Service $06 – Request on–board monitoring test result for non–continuosly monitored system

This service is available and totally compatible with Standard 15031–5 only for CCM.

12.9

5.7. Service $07 – Request on–board monitoring test result for continuosly monitored system

This service is available and totally compatible with Standard 15031–5.

12.10

5.8. Service $08 – Request control of on–board system, test or components

This service is available and totally compatible with Standard 15031–5.

12.11

5.9. Service $09 – Request vehicle information

This service is available and totally compatible with Standard 15031–5; the required PIDs are quoted on ANNEX E (CCM – TCM/CAE)

Annex A
PID (Parameter ID)/TID (Test ID)/INFOTYPE supported definition

The PID $00 must be available for the following services: $01, $02, $05, $06, $08 and $09 and totally compatible with Standard ISO 15031−5.

Annex D (CCM)
TIDs (Test ID) for service $08 scaling and definition

It must not be improved to a value less than TID 0.

Annex E (CCM – TCM/CAE)
INFOTYPEs for service $09 scaling and definition

The PID hereinafter listed are compulsory:

Info Type (Hex)	Vehicle information data byte description	Scaling	Mnemonic
03	**MessageCount CALID**	1 byte unsigned	MC_CALID
04	**Calibration Identifications**	16 ASCII characters	CALID

Annex B (TCM/CAE)
PIDs (Parameter ID) for service $01 scaling and definition

The PID hereinafter listed are compulsory for the applications (TCM/CAE); for other applications, all available parameters shall be accessible.

Info Type (Hex)	Definition
01	Number of emission−related powertrain DTCs and MI status − On−board diagnostic evaluation − Continuous monitoring tests − Supported tests run at least once per trip − Status of tests run at least once per trip
0C	Engine speed
0D	Vehicle speed
11	Absolute throttle position