

# **ISO-14229**

**Road vehicles-Unified diagnostic services (UDS)-  
Specification and requirements**

# 目 录

前言 .....	4
介绍 .....	4
1 适用范围.....	5
2 参考标准.....	6
3 术语及定义.....	6
3.1 INTEGER TYPE .....	6
3.2 DIAGNOSTIC TROUBLE CODE .....	6
3.3 DIAGNOSTIC SERVICE.....	6
3.4 CLIENT .....	6
3.5 SERVER .....	6
3.6 TESTER.....	6
3.7 DIAGNOSTIC DATA .....	6
3.8 DIAGNOSTIC SESSION .....	7
3.9 DIAGNOSTIC ROUTINE.....	7
3.10 RECORD .....	7
3.11 SECURITY .....	7
3.12 FUNCTIONAL UNIT .....	7
3.13 LOCAL SERVER .....	7
3.14 LOCAL CLIENT.....	7
3.15 REMOTE SERVER .....	7
3.16 REMOTE CLIENT .....	8
3.17 PERMANENT DTC .....	8
4 符号及缩写术语.....	8
5 一般性要求.....	8
6 应用层服务.....	8
7 应用层标准.....	8
8 服务描述的一般性要求.....	8
9 诊断和通信管理功能单元.....	8
10 数据传输功能单元.....	8
10.1 概述 .....	8
10.2 READDATAByIDENTIFIER (22 HEX) 服务 .....	9
10.2.1 服务描述 .....	9
10.2.2 需求报文 .....	9
10.2.2.1 需求报文定义 .....	9
10.2.2.2 需求报文子功能参数\$Level(LEV_)定义 .....	9
10.2.2.3 需求报文数据参数定义.....	9
10.2.3 正向响应报文.....	10
10.2.3.1 正向响应报文定义.....	10
10.2.3.2 正向响应报文数据参数定义.....	10

10.2.4	所支持的负向响应编码.....	10
10.2.5	ReadDataByIdentifier 服务报文流程举例.....	11
10.2.5.1	假设 .....	11
10.2.5.2	例 #1— 读取单一数据标识符 F190 hex (VIN number: vehicle identification number) .....	12
10.2.5.3	例 #2 ——读取多重数据标识符 010A hex 和 0110 hex .....	13
10.3	READMEMORYBYADDRESS (23 HEX)服务 .....	13
10.3.1	服务描述 .....	13
10.3.2	需求报文 .....	14
10.3.2.1	需求报文定义 .....	14
10.3.2.2	需求报文子功能参数 \$Level(LEV_)定义 .....	14
10.3.2.3	需求报文数据参数定义.....	14
10.3.3	正向响应报文.....	15
10.3.3.1	正向响应报文定 .....	15
11	存储数据发送功能单元.....	16
12	输入输出控制功能单元.....	16
13	远程唤醒例程功能单元.....	16
14	上传及下载功能单元.....	16
附录 A	全局参数定义 .....	16
附录 B	诊断及通信管理功能模块数据参数定义 .....	16
附录 C	数据传输功能单元数据参数定义.....	17
附录 D	存储数据传输功能单元数据参数定义.....	17
附录 E	输入输出控制功能单元数据参数定义.....	17
附录 F	例程远程激活功能单元数据参数定义 .....	17
附录 G	ADDRESSANDLENGTHFORMATIDENTIFIER 参数值举例.....	17

# 前言

ISO 是一个国际标准团体的世界范围的联盟。通常，ISO 技术会议的主要工作就是准备国际性标准。

# 介绍

ISO14229 标准是为了给诊断系统定义通用的需求而发布的，而这与串行数据链路的种类无关。

为了完成这一工作，该标准是基于 OSI 基准模式，这与 ISO7498-1 和 ISO/IEC 10731 保持一致的，在这两个标准中，它把通讯系统的结构分为七层。当映射到这种模式时，可以通过诊断仪（diagnostic tester/client）和电子控制单元（ECU，server）来将服务分解为两种：

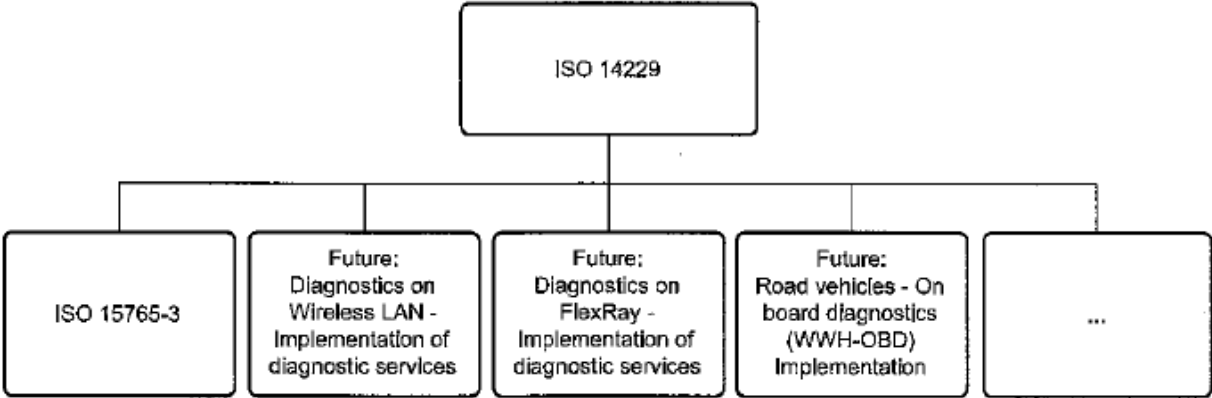
- 单独的诊断服务（第七层），和
- 通信服务（第一到第六层）。

Note: ISO14229 中的诊断服务可以在不同的应用里执行，比如 ISO16844（全部），ISO11992（全部），ISO9141（全部），以及 ISO14230（全部）等。

Applicability	OSI layer	Enhanced diagnostics services (non-emissions-related)	
Seven layers according to ISO/IEC 7498-1 and ISO/IEC 10731	Application (layer 7)	ISO 14229/ISO 15765-3/ISO 11992-4	ISO 14229/further standards
	Presentation (layer 6)	—	—
	Session (layer 5)	ISO 15765-3/ISO 11992-4	further standards
	Transport (layer 4)	ISO 15765-2/ISO 11992-4	further standards
	Network (layer 3)	ISO 15765-2/ISO 11992-4	further standards
	Data link (layer 2)	ISO 11898/ISO 11992-1/SAE J1939-15	further standards
	Physical (layer 1)	ISO 11898/ISO 11992-1/SAE J1939-15	further standards

表 1-符合 OSI 层的诊断/编程举例

下面的图 1 就是一个将来在不同数据链路层之上对 ISO14229 的一个扩张的举例。



## 统一诊断服务（UDS）——定义及需求

### 1 适用范围

ISO14229 标准定义了诊断服务的数据链路的独立需求，这些诊断服务允许诊断仪（client 客户端）在车载电子控制单元（server 服务器）里控制诊断功能，这些功能包括电子燃油喷射、自动变速箱、ABS 系统能，而这些控制单元都通过串行数据链路被嵌入到整车上。该标准指定了一些通用服务，这些服务允许诊断仪（client）停止或者恢复数据链路上的非诊断报文的传输。ISO14229 标准不适用于非诊断报文的传输以及在两个电子控制单元之间进行通信。它不指定任何执行请求。

基于 ISO14229 标准的整车诊断构架支持：

- 单独一个的检测仪（client）临时性的或者永久的连接到车载诊断数据链上；并且
- 多个车载电子控制单元（server）直接或者间接相连。

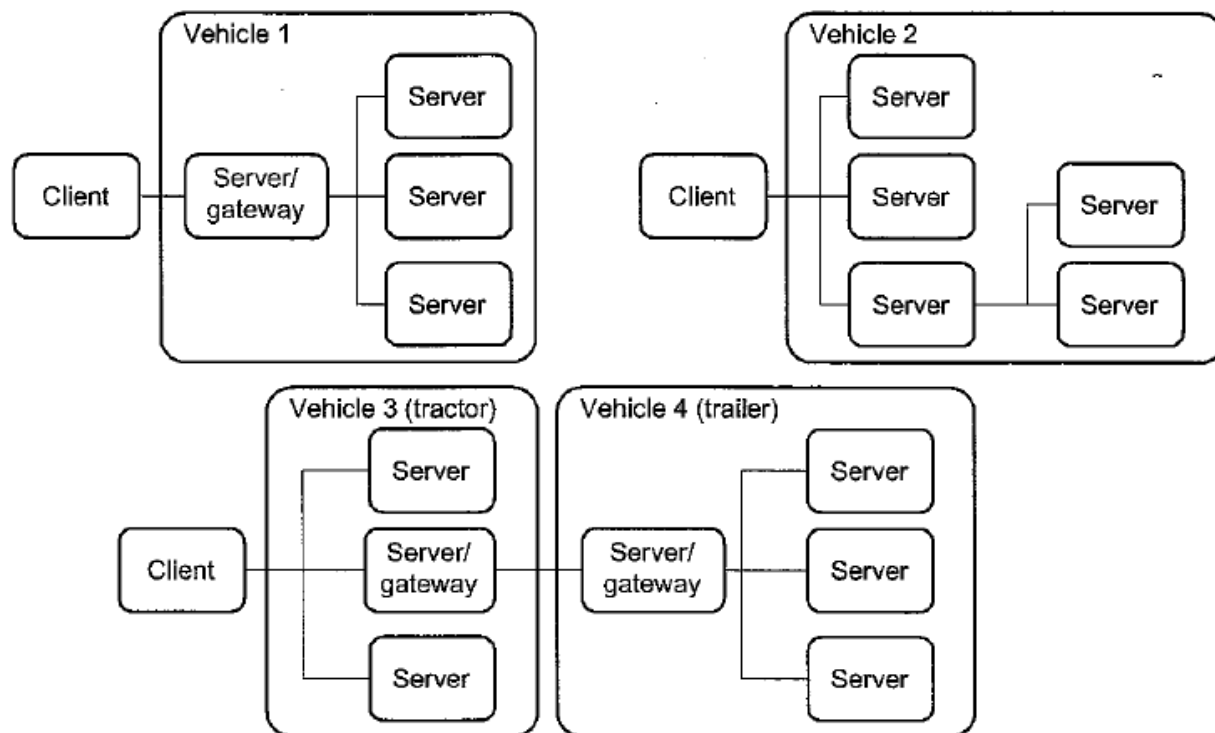


图 2-车辆诊断系统构架

图 2 中，

- 对于车辆 1，各服务器都是通过内部数据链路相连的，而与诊断数据链路都是非直接连接，中间通过了一个网关。ISO14229 支持诊断数据链路路上的诊断通信，对于内部数据链路路上的诊断通信可以参考 ISO14229 或者其他标准。
- 对于车辆 2，各服务器都直接与诊断数据链路相连。
- 对于车辆 3，各服务器的连接方式与车辆 2 的是一样的，并且车辆 4 将它的网关与车辆 3 的网关直接相连。

## 2 参考标准

## 3 术语及定义

以下的数据机定义适用于整个文档。

### 3.1 Integer type

### 3.2 Diagnostic trouble code

故障码：由 OBD 系统定义的，针对某种故障的通用的数字标识符。

### 3.3 Diagnostic service

诊断服务：客户端为了从服务器获取故障信息，或者处于故障诊断目的而去调整服务器的动作，而发起的信息交换。

### 3.4 Client

客户（客户端）：其功能是 tester 的一部分，并且使用诊断服务。

Note：通常，tester 使用其它功能，例如数据库管理、细节解释以及人机接口等。

### 3.5 Server

服务器：其功能是电子控制单元的一部分，并且提供诊断服务。

Note：ISO14229 之所以区分 server 和电子控制单元，是因为该国际标准与执行是相独立的。

### 3.6 Tester

检测仪：作为系统，其控制功能包括检测、测试、车载电子控制单元的监控或者诊断，而这些控制单元都应专用于特定类型的操作者（例如一种修车间技工专用的扫描仪或者装配车间专用的检测工具）。

Note：检测仪也指客户端。

### 3.7 Diagnostic data

该数据是被保存到电子控制单元的内存中，它可以通过检测仪来检测或者修改（诊断数据包括模拟输入输出、数字输入输出、中间值及不同状态信息）。

举例

诊断数据的例子包括车速、节气门转角、反光镜位置、系统状态等等。这里定义了三

种不同类型的诊断数据：

——当前值：由电子控制单元的正常操作所使用或者产生的当前的值；

——存储值：它是在特定时刻产生的一个当前值的内部拷贝，例如，当有故障发生或者周期产生时（该拷贝值的产生是在电子控制单元的控制之下进行的）。

——静态值：例如 VIN；服务器没有被强制要求为诊断目的而保存内部拷贝值，在这种情况下检测仪应改仅仅需要当前值。

### **3.8 Diagnostic session**

指服务器的当前模式，它影响诊断功能的级别。

Note：定义一个维修车间或者开发检测 session 会选择不同的服务器功能（比如进入所有内存区域只在开发检测 session 时才被允许）。

### **3.9 Diagnostic routine**

诊断程序是嵌入到控制单元内的程序，它可以由服务器经由客户端发出一个请求来触发。

Note：它既可以代替正常模式程序运行，也可以同时运行正常模式程序。在第一种情况下，ECU 不可能进行正常操作。在第二种情况下，可以使用多重诊断程序，这时电子控制单元的其他部分的功能是正常的。

### **3.10 Record**

记录是指一个或者多个诊断数据组分。

### **3.11 Security**

安全，在 ISO14229 中用到的安全是指一种安全通过的办法，这种方法满足篡改保护的需求，该保护功能是在 ISO15031-7 中定义的。

### **3.12 Functional unit**

功能单元：是指一组功能上接近或者具有补充作用的诊断服务。

### **3.13 Local server**

局部服务器，是指与客户端连接到相同的局域网，并且是与客户端有相同地址空间的服务器的一部分。

### **3.14 Local client**

局部客户端，是指与服务器连接到相同的局域网，并且是与服务器有相同地址空间的客户端的一部分。

### **3.15 Remote server**

远程服务器，是指不直接连接到主诊断网络的服务器。

Note 1：远程服务器是由远程网络地址来定义的。远程网络地址用以表示一个自己的网络地址空间，该地址是与主网络地址相独立的。

Note 2：通过主网络上的局部客户端可以连接远程服务器。主网络上的每个局部服务器都能作为一个通往一组独立远程服务器的通道。所以，通常用一对地址来标识一个远程服务

器：一个局部地址用来标识通往远程服网络的通道，而另一个远程地址用来标识远程服务器自己。

3.16 Remote client

远程客户端，是指没有直接连接到主诊断网络的客户端。

Note：一个远程客户端是由远程网络地址来定义的。远程网络地址表示一个自己的地址空间，该地址是与主网络地址相独立的。

3.17 Permanent DTC

永久故障码，永久故障码被存储在 NVRAM 中，并且不能因任何检测设备命令或者车载电脑断电而清除。

4 符号及缩写术语

5 一般性要求

6 应用层服务

7 应用层标准

8 服务描述的一般性要求

9 诊断和通信管理功能单元

10 数据传输功能单元

10.1 概述

服务	描述
ReadDataByIdentifier	客户端要求读取一个记录的当前值，该记录是由一个给定的数据标识符标识的。
ReadMemoryByAddress	客户端要求读取所提供的存储器范围内的当前值。
ReadScalingDataByIdentifier	客户端要求读取一个记录的缩放信息，该记录是由一个给定的数据标识符标识的。
ReadDataByPeriodicIdentifier	客户端要求在周期性传送服务中安排数据。
DynamicallyDefineDataIdentifier	客户端要求动态定义数据标识符，这些标识符可以由随后的 ReadDataByIdentifier 服务来读取。



WriteDataByIdentifier	客户端要求写入一个由给定数据标识符指定的记录。
WriteMemoryByAddress	客户端要求在给定寄存器范围内写入。

表 128-数据传输功能单元

10.2 ReadDataByIdentifier（22 hex）服务

10.2.1 服务描述

ReadDataByIdentifier 服务允许客户端从由一个或者多个数据标识符标识的服务器中请求数据记录。

客户端请求报文包括一个 2byte 的数据标识符值，该标识符值指定了由服务器支持的数据记录（关于所允许的标识符的值，具体参见 C.1）。该数据记录的格式和定义因该有整车厂或者系统供应商指定，并且，如果由服务器支持的话，还应该包括模拟输入输出信号、数字输入输出信号、内部数据以及系统状态信息。

服务器应该限制数据标识符的数量，以使它们能同时被整车厂以及系统供应商所接受。

注意-服务器和客户端必须满足 7.5.3 中定义的要求以及响应报文动作，事件中的那些寻址方法都是为该服务而执行的。

10.2.2 需求报文

10.2.2.1 需求报文定义

A_Data byte	Parameter name	Cvt	Hex value	Mnemonic
#1	ReadDataByIdentifier Request Service Id	M	22	RDBI
#2	dataIdentifier[] #1 = [ byte#1 (MSB) byte#2 ]	M	00-FF	DID_ HB
#3		M	00-FF	LB
:	:	:	:	:
#n-1	dataIdentifier[] #m = [ byte#1 (MSB) byte#2 ]	U	00-FF	DID_ HB
#n		U	00-FF	LB

表-129 需求报文定义

10.2.2.2 需求报文字功能参数\$Level(LEV\_)定义

该服务没有使用子功能参数。

10.2.2.3 需求报文数据参数定义

下面的数据参数都是为该服务定义的：

Definition
<b>dataIdentifier (#1 to #m)</b> This parameter identifies the server data record(s) being requested by the client (see C.1 for detailed parameter definition).

表 130-需求报文数据参数定义

## 10.2.3 正向响应报文

### 10.2.3.1 正向响应报文定义

A_Data byte	Parameter name	Cvt	Hex value	Mnemonic
#1	ReadDataByIdentifier Response Service Id	M	62	RDBIPR
#2	dataIdentifier[] #1 = [ byte#1 (MSB) byte#2 ]	M	00-FF	DID_ HB
#3		M	00-FF	LB
#4	dataRecord[] #1 = [ data#1 : data#k ]	M	00-FF	DREC_ DATA_1
:		:	:	:
#{k-1}+4		U	00-FF	DATA_m
:	:	:	:	:
#n-(o-1)-2	dataIdentifier[] #m = [ byte#1 (MSB) byte#2 ]	U	00-FF	DID_ HB
#n-(o-1)-1		U	00-FF	LB
#n-(o-1)	dataRecord[] #m = [ data#1 : data#o ]	U	00-FF	DREC_ DATA_1
:		:	:	:
#n		U	00-FF	DATA_k

表-131 正向响应报文定义

### 10.2.3.2 正向响应报文数据参数定义

定义
<b>dataIdentifier(#1 to #m)</b> 该参数是一个来自请求报文的针对数据参数标识符的回应。
<b>dataRecord(#1 to #k/o)</b> ReadDataByIdentifier 正向响应报文使用该参数为客户端提供所需要的数据记录值。数据记录的内容不在该文档中定义，它属于整车厂定义范围。

表 132-响应报文数据参数定义

## 10.2.4 所支持的负向响应编码

该服务应该执行下列的负向响应代码。每一种响应编码所发生的环境如表-133 所示。

Hex	描述	Cvt	Mnemonic
13	<b>IncorrectMessageLengthOrInvalidFormat</b> 如果需求报文的长度是无效的，那么将发送该响应编码。	M	IMLOIF
22	<b>conditionsNotCorrect</b> 如果针对完成该需求动作的服务器的造作环境都不满足，那么那	U	CNC

	么将发送该响应编码。		
31	<b>requestOutOfRange</b> 下面两种情况下，该编码将被发送： 1、所有的需求数据标识符值都不被设备所支持。 2、客户端超出了在一次请求中被允许的最大数据标识符的数量。	M	ROOR
33	<b>securityAccessDenied</b> 如果至少有一个数据标识符被保护，并且服务器没有处在非锁定状态，那么该编码将被发送。	M	SAD

表-133 负向响应编码

10.2.5 ReadDataByIdentifier 服务报文流程举例

10.2.5.1 假设

该部分内容主要是设定了一些条件，只有在满足这些条件的基础上，才能实现 ReadDataByIdentifier 服务的例子。

下面的数据标识符的例子都是指动力传动系统设备（如发动机控制模块）。关于排放相关的一些专用词、定义以及缩写等的详细信息，可以参考 ISO/TR15031-2 的相关内容。

第一个例子是，读取一个 2byte 的数据标识符，它包含了一个单片信息（其中标识符 F190 hex 包括了 VIN 数）。

第二个例子列举了用一个单一请求来实现多重数据标识请求（其中标识符 010A hex 包括发动机冷却液温度、节气门位置、发动机转速、进气歧管绝对压力、空气流量、车速传感器、大气压力、计算负载值、怠速空气控制以及加速踏板位置信号，而数据标识符 0110hex 包括蓄电池正极电压）。

### 10.2.5.2 例 #1— 读取单一数据标识符 F190 hex （VIN number: vehicle identification number）

**Table 134 — ReadDataByIdentifier request message flow example #1**

<b>Message direction:</b>	client → server		
<b>Message type:</b>	Request		
A_Data byte	Description (all values are in hexadecimal)	Byte value (hex)	Mnemonic
#1	ReadDataByIdentifier request SID	22	RDBI
#2	dataIdentifier [ byte#1 ] (MSB)	F1	DID_B1
#3	dataIdentifier [ byte#2 ]	90	DID_B2

**Table 135 — ReadDataByIdentifier positive response message flow example #1**

<b>Message direction:</b>	server → client		
<b>Message type:</b>	Response		
A_Data byte	Description (all values are in hexadecimal)	Byte value (hex)	Mnemonic
#1	ReadDataByIdentifier response SID	62	RDBIPR
#2	dataIdentifier [ byte#1 ] (MSB)	F1	DID_B1
#3	dataIdentifier [ byte#2 ]	90	DID_B2
#4	dataRecord [ data#1 ] = VIN Digit 1 = "W"	57	DREC_DATA1
#5	dataRecord [ data#2 ] = VIN Digit 2 = "0"	30	DREC_DATA2
#6	dataRecord [ data#3 ] = VIN Digit 3 = "L"	4C	DREC_DATA3
#7	dataRecord [ data#4 ] = VIN Digit 4 = "0"	30	DREC_DATA4
#8	dataRecord [ data#5 ] = VIN Digit 5 = "0"	30	DREC_DATA5
#9	dataRecord [ data#6 ] = VIN Digit 6 = "0"	30	DREC_DATA6
#10	dataRecord [ data#7 ] = VIN Digit 7 = "0"	30	DREC_DATA7
#11	dataRecord [ data#8 ] = VIN Digit 8 = "4"	34	DREC_DATA8
#12	dataRecord [ data#9 ] = VIN Digit 9 = "3"	33	DREC_DATA9
#13	dataRecord [ data#10 ] = VIN Digit 10 = "M"	4D	DREC_DATA10
#14	dataRecord [ data#11 ] = VIN Digit 11 = "B"	42	DREC_DATA11
#15	dataRecord [ data#12 ] = VIN Digit 12 = "5"	35	DREC_DATA12
#16	dataRecord [ data#13 ] = VIN Digit 13 = "4"	34	DREC_DATA13
#17	dataRecord [ data#14 ] = VIN Digit 14 = "1"	31	DREC_DATA14
#18	dataRecord [ data#15 ] = VIN Digit 15 = "3"	33	DREC_DATA15
#19	dataRecord [ data#16 ] = VIN Digit 16 = "2"	32	DREC_DATA16
#20	dataRecord [ data#17 ] = VIN Digit 17 = "6"	36	DREC_DATA17

### 10.2.5.3 例 #2 ——读取多重数据标识符 010A hex 和 0110 hex

**Table 136 — ReadDataByIdentifier request message flow example #2**

<b>Message direction:</b>		client → server	
<b>Message type:</b>		Request	
A_Data byte	Description (all values are in hexadecimal)	Byte value (hex)	Mnemonic
#1	ReadDataByIdentifier request SID	22	RDBI
#2	dataIdentifier #1 [ byte#1 ] (MSB)	01	DID_B1
#3	dataIdentifier #1 [ byte#2 ]	0A	DID_B2
#4	dataIdentifier #2 [ byte#1 ] (MSB)	01	DID_B1
#5	dataIdentifier #2 [ byte#2 ]	10	DID_B2

**Table 137 — ReadDataByIdentifier positive response message flow example #2**

<b>Message direction:</b>		server → client	
<b>Message type:</b>		Response	
A_Data byte	Description (all values are in hexadecimal)	Byte value (hex)	Mnemonic
#1	ReadDataByIdentifier response SID	62	RDBIPR
#2	dataIdentifier [ byte#1 ] (MSB)	01	DID_B1
#3	dataIdentifier [ byte#2 ] (LSB)	0A	DID_B2
#4	dataRecord [ data#1 ] = ECT	A6	DREC_DATA1
#5	dataRecord [ data#2 ] = TP	66	DREC_DATA2
#6	dataRecord [ data#3 ] = RPM	07	DREC_DATA3
#7	dataRecord [ data#4 ] = RPM	50	DREC_DATA4
#8	dataRecord [ data#5 ] = MAP	20	DREC_DATA5
#9	dataRecord [ data#6 ] = MAF	1A	DREC_DATA6
#10	dataRecord [ data#7 ] = VSS	00	DREC_DATA7
#11	dataRecord [ data#8 ] = BARO	63	DREC_DATA8
#12	dataRecord [ data#9 ] = LOAD	4A	DREC_DATA9
#13	dataRecord [ data#10 ] = IAC	82	DREC_DATA10
#14	dataRecord [ data#11 ] = APP	7E	DREC_DATA11
#15	dataIdentifier [ byte#1 ] (MSB)	01	DID_B1
#16	dataIdentifier [ byte#2 ] (LSB)	10	DID_B2
#17	dataRecord [ data#1 ] = B+	8C	DREC_DATA1

## 10.3 ReadMemoryByAddress (23 hex)服务

### 10.3.1 服务描述

ReadMemoryByAddress 服务允许客户端通过一个提供的起始地址从服务器中请求寄存器数据，以及指定所读取数据的大小。

ReadMemoryByAddress 请求报文被用与从服务器中读取寄存器数据，该服务器是由 memoryAdress 和 memorySize 这两个参数定义的。其中，memoryAddress 指定了 byte 的个数，而 memorySize 是由 addressAndLengthFormatidentifier（高低字节）来定义的。

同样，也可以使用固定的 addressAndLengthFormatidentifier 和参数 memoryAddress 及 memorySize 的高字节范围中没有使用的位，这些位都以 00 hex 来填充。

为了防止内存地址的重叠，也可以只用附加的 memoryAddress 字节作为内存标识符（比如适用内部或者外部的 Flash）。

服务器通过 ReadMemoryByAddress 的正向响应报文来发送数据记录的值。数据记录参数的格式及定义应该由整车厂来最终确定。数据记录参数应该包括模拟输入输出信号、数字输入输出信号、内部数据以及系统的状态信息（如果服务器支持的话）。

IMPORTANT-服务器和客户端因该就请求和响应报文及动作达成一致，正如 7.5.3 中定义的那样，该事件中那些寻址方法都是为这一服务而执行的。

10.3.2 需求报文

10.3.2.1 需求报文定义

A_Data byte	Parameter name	Cvt	Hex value	Mnemonic
#1	ReadMemoryByAddress Request Service Id	M	23	RMBA
#2	addressAndLengthFormatIdentifier	M	00-FF	ALFID
#3 : #(m-1)+3	memoryAddress[] = [ byte#1 (MSB) : byte#m ]	M : C <sub>1</sub> <sup>a</sup>	00-FF : 00-FF	MA_ B1 : Bm
#n-(k-1) : #n	memorySize[] = [ byte#1 (MSB) : byte#k ]	M : C <sub>2</sub> <sup>b</sup>	00-FF : 00-FF	MS_ B1 : Bk
<sup>a</sup> The presence of the C <sub>1</sub> parameter depends on address length information parameter of the addressAndLengthFormatIdentifier.				
<sup>b</sup> The presence of the C <sub>2</sub> parameter depends on the memory size length information of the addressAndLengthFormatIdentifier.				

表 138-需求报文定义

10.3.2.2 需求报文字功能参数 \$Level(LEV\_)定义

该服务不使用子功能参数。

10.3.2.3 需求报文数据参数定义

下面的数据参数都是为该服务定义的。

定义
<b>addressAndLengthFormatIdentifier</b> 该参数是一个字节的数值，而其每个半字节都是被单独编码的（见附录 G.1 的举例）： Bit7-4: 参数 <b>memorySize</b> 的长度（字节数）。 Bit3-0: 参数 <b>memoryAddress</b> 的长度（字节数）。
<b>memoryAddress</b> 该参数是服务器中重新取回数据的起始地址。该地址（即该参数）的字节长度是用 <b>addressFormatIdentifier</b> 的低四位（bit3-0）定义的。参数 <b>memoryAddress</b> 中的#m 字节常常是服务器中被引用的地址的最低有效位（LSB）。而该地址的最高有效位（MSB）可以被用作寄存器标识符。 一个使用寄存器标识符的例子可以是一个带有 16 位寻址及寄存器地址重用（即，对于服务器来讲，它要使用一个给定的有效地址就必须另外带一个物理寄存器设备，而对于内部或者外部的 Flash，则可以直接使用这样的给定有效地址）的双处理器服务器。在这种情况下，参数 <b>memoryAddress</b> 内的另一个没有使用的字节就可以被定义为一个寄存器标识符，用以选择所希望的寄存器设备。这种功能的应用应该由整车厂或者系统供应商来定义。
<b>memorySize</b> 该参数属于 <b>ReadMemoryByAddress</b> 需求报文，它定义了从指定地址中将被读取的字节数，而这些地址是由服务器寄存器中的 <b>memoryAddress</b> 来定义的。而该字节数本身（即该参数）所占的大小是由 <b>addressFormatIdentifier</b> 的高四位（bit7-4）定义的。

10.3.3 正向响应报文

10.3.3.1 正向响应报文定义

A_Data byte	Parameter name	Cvt	Hex value	Mnemonic
#1	ReadMemoryByAddress Response Service Id	M	63	RMBAPR
#2	dataRecord[] = [ data#1 : data#m ]	M	00-FF	DREC_ DATA_1
:		:	:	:
#n		U	00-FF	DATA_m

表 140-正向响应报文定义

10.3.3.2 正向响应报文数据参数定义

定义
<b>dataRecord</b> <b>ReadMemoryByAddress</b> 的正向响应报文使用该参数为客户端提供所需要的数据记录值。 <b>dataRecord</b> 的内容并没有在本标准里定义，而且该内容应该反映所需求寄存器内容。数据格式应该由整车厂或者系统供应商定义。

10.3.4 所支持的负向响应编码（NRC\_）

下列的负向响应编码应该为该服务所执行。各响应编码发生的环境如下表 142 所示。

Hex	描述	Cvt	Mnemonic
13	incorrectMessageLengthOrInvalidFormat 报文长度有错。		
22	conditionsNotCorrect 如果服务器的操作条件不足以完成所请求的动作，那么该响应编码将被发送。		
31	requestOutOfRange 该响应编码在下列情况下发出： 1、在[\$MA, (\$MA+\$MS-\$1)]间隔内的任何寄存器地址都是无效的。 2、在[\$MA, (\$MA+\$MS-\$1)]间隔内的任何寄存器地址都是受限制的。 3、需求报文中的 memorySize 参数值要比服务器所支持的最大值要大。 4、定义的 addressAndLengthFormatIdentifier 是无效的。		
33	SecurityAccessDenied 如果在[\$MA, (\$MA+\$MS-\$1)]间隔中的任何寄存器地址都是可靠的，而且服务器处于锁止状态，那么将发送该编码。		

10.3.5 ReadMemoryByAddress 的报文流程举例

10.3.5.1 假设

11 存储数据发送功能单元

12 输入输出控制功能单元

13 远程唤醒例程功能单元

14 上传及下载功能单元

附录 A 全局参数定义

附录 B 诊断及通信管理功能模块数据参数定义



附录 C 数据传输功能单元数据参数定义

附录 D 存储数据传输功能单元数据参数定义

附录 E 输入输出控制功能单元数据参数定义

附录 F 例程远程激活功能单元数据参数定义

附录 G **addressAndLengthFormatIdentifier** 参数值举例