## OBD和UDS服务的区别

上一讲讲了UDS服务，对于现在大部分汽车工程师可能接触到地都是UDS服务，在大部分的汽车电子ECU上，只需要UDS服务就足够了。

OBD(On-Board-Diagnose)车载诊断，该服务也是DCM模块支持的一种服务，主要是为了监控排放相关的系统，比如发送机和变速箱。

OBD服务定义了排放相关系统必须支持的诊断服务和数据格式，OBD服务底层数据传输可以走K线也可以走CAN线，现在大部分都是走CAN线。OBD和UDS是并列的一套应用层协议，对于与排放相关的ECU，通常既要支持UDS协议也要支持OBD协议。

OBD协议是由ISO-15031规定，我们主要关注ISO-15031-5，关注OBD支持的具体服务即可。

**注意：其实OBD服务都可以通过UDS服务来实现，相信未来UDS服务会取代掉OBD服务，称为真正的统一诊断服务。**

## OBD有哪些服务

OBD支持0x01-0x0A共10个服务。0x01-0x0A就是对应的服务ID即SID。

|  |  |
| --- | --- |
| 服务  （16进制） | 描述 |
| 01 | Request current powertrain diagnostic data |
| 请求当前动力总成诊断数据 |
| 02 | Request powertrain freeze frame data |
| 请求动力系统冻结帧数据 |
| 03 | Request emission-related diagnostic trouble codes |
| 请求与排放相关的诊断故障代码 |
| 04 | Clear/Reset emission-related diagnostic information |
| 请求/重置排放相关诊断信息 |
| 05 | Request oxygen sensor monitoring test results |
| 请求氢气传感器监测测试结果 |
| 06 | Request On-board monitoring test results for specific monitored systems |
| 请求特定监控系统的车载测试监控结果 |
| 07 | Request emission-related diagnostic trouble codes detected during current or last completed driving cycle |
| 请求在当前或上次完成的驾驶循环中检测到的与排放相关的诊断故障代码 |
| 08 | Request control of on-board system,test,or component |
| 请求控制车载系统、测试或组件 |
| 09 | Request vehicle information |
| 请求车辆信息 |
| 0A | Request emissions-related diagnostic trouble codes with permanent status |
| 请求具有永久状态的与排放相关的诊断故障代码 |

Service 01 – Request Current Powertrain Diagnostic Data:

该服务用于读取动力系统当前的诊断数据，比如某个传感器的状态、发动机转速、DTC数量、故障指示灯是否亮起等，命令格式是SID+若干PID（Parameter ID）。每个PID也是一个byte，所以理论上PID取值范围是0x00至0xFF，但是ISO-15031-5只明确定义了部分PID，其余的值都保留。问题来了，OBD定义了如此多的PID，那么某个ECU到底支持哪些PID，诊断仪是如何获知的呢？

实际上，PID分为两类，

一类用于具体的PID数据

而另外一类则用来读取ECU支持哪些PID

用于第二种目的的PID分别是0x00，0x20，0x40…，读取其中一个ID后ECU会返回4个字节的结果，这4个字节中的每个bit表示其所对应的PID是否被支持。以下面这个例子来说明就很容易理解了：

服务ID为01 PID为00 00是用来读取该ECU支持哪些PID的

|  |  |  |
| --- | --- | --- |
| **Data Byte** | **Description(All PID values are in hexadecimal)** | **Byte Values(Hex)** |
| #1 | Request current powertrain diagnostic data request SID | 01 |
| #2 | PID used to determine PID support for PIDs 01-20 | 00 |

诊断应答

|  |  |  |
| --- | --- | --- |
| Data Byte | **Description(All PID values are in hexadecimal)** | **Byte Values(Hex)** |
| #1 | Request current powertrain diagnostic data request SID | 41 |
| #2 | PID requested | 00 |
| #3 | Data byte A.representing support for PIDs 01,03-08 | 10111111b=$BF |
| #4 | Data byte B.representing support for PIDs 09,0B-10 | 10111111b=$BF |
| #5 | Data byte C.representing support for PIDs 11,13-15 | 10101000b=$A8 |
| #6 | Data byte D.representing support for PIDs 19,1C-20 | 10010001b=$91 |

正响应诊断应答服务号也是0x01 + 0x40，第二个byte为PID 0x00，后面4个byte读出当前ECU支持哪些PID，比如10111111 从高位开始，为1的位代表支持对应PID，最高位为1，支持0x01，不支持0x02，支持0x03-0x08.

诊断仪通过0x00 0x20 0x40这些PID，可以读取当前ECU支持哪些PID

**Service 02 – Request Powertrain Freeze Frame Data**

一旦ECU确定了某个故障，就要把这个故障被确定时地相关状态信息“冻结”下来，即所谓地冻结帧，这些状态信息对车辆故障地确定非常重要，因为它们记录了车辆发生故障时地很多相关信息，这些状态信息数据必须在ISO-15031-5的PID列表中选择（与Service 01使用的PID列表相同）。02命令和01命令的使用方式非常相似，只不过02读取的是故障发生时的数据，而01读取的当前数据，数据格式和含义都是相同的。与01命令不同的是，02命令中多了1个frame字节，如下图所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Byte** | **Parameter Name** | **Cvt** | **New Value** |
| **#1** | **Request powertrain freeze data request SID** | **M** | **02** |
| #2 | PID#1(see Appendix B) | M/C1 | xx |
| #3 | Frame# | M | xx |
| #4 | PID#2(see Appendix B) | U/C1 | xx |
| #5 | Frame# | C2 | xx |
| #6 | PID#3(see Appendix B) | U/C1 | xx |
| #7 | Frame# | C2 | xx |

OBD规定，用frame = 0x00来代表读取冻结帧。如果主机厂想自己再定义些什么其他的帧，或者多定义几个冻结帧，则可以给frame分配上其他的编号。

需要指出的是，OBD只规定了ECU需要为一个DTC存储冻结帧，当ECU中同时存在多个DTC时，就要根据优先级来判定存储谁的冻结帧了。

**Service 03 – Request Emission-Related Diagnostic Trouble Codes**

服务03用于读取存储在ECU中的与排放相关的“confirmed”DTC,用法非常简单，它没有任何参数，诊断仪只需要发送03就可以。下面两张图分别展示了03命令的请求的响应格式。

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Byte** | **Parameter Name** | **Cvt** | **New Value** |
| **#1** | **Request emission-related DTC request SID** | **M** | **03** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Byte** | **Parameter Name** | **Cvt** | **New Value** |
| **#1** | **Request emission-related DTC response SID** | **M** | **43** |
| #2 | #cf DTC = [no emission-related DTCs stored emission-related DTCs stored] | M | Xx = 1  00  01-FF |
| #3 #4 | DTC#1(High Byte)  DTC#1(Low Byte) | C  C | Xx  xx |
| … |  | … | xx |
| #n-1  #n | DTC#n(High Byte)  DTC#n(Low Byte) | C  C | Xx  xx |

在03命令的响应中，第2个字节表示DTC数量，后面每两个字节表示一个DTC。冻结帧可以理解为快照。

**Service 04 – Clear/Reset Emission-Related Diagnostic Information**

04服务用于清空ECU中存储的与排放相关的DTC，除了DTC以外，以下的信息也要被清除。

|  |  |
| --- | --- |
| Number of diagnostic trouble codes | (can be read with Service $01,PID $01) |
| Diagnostic trouble codes | (can be read with Service $03) |
| Trouble code for freeze frame data | (can be read with Service $02,PID $02) |
| Freeze frame data | (can be read with Service $02) |
| Status of system monitoring tests | (can be read with Service $01,PID $01) |
| On-board monitoring test results | (can be read with Service $06 and $07) |
| Distance travelled while MIL is activated | (can be read with Service $01,PID $21) |
| Number of warm-ups since DTC cleared | (can be read with Service $01,PID $30) |
| Distance since diagnostic trouble codes cleared | (can be read with Service $01,PID $31) |
| Minutes run by the engine while MIL activated | (can be read with Service $01,PID $4D) |
| Time since diagnostic trouble codes cleared | (can be read with Service $01,PID $4E) |

执行04命令时被清理的信息

它的使用非常简单，请求是一个字节的04，响应是一个字节的44。只有在发动机没有运转的时候才可以执行这个服务，否则ECU应该给出NRC 0x22（条件步满足）来拒绝该服务

**Service 05 – Request Oxygen Sensor Monitoring Test Results**

05服务用于读取氧传感器的状态，对于OBDonCAN来说不支持该服务，相应的功能由06服务实现。

**Service 06 – Request On-Board Monitoring Test Results for Specific Monitored Systems**

该服务用于请求对特定被监测系统的监测结果。OBD中定义了一个MID（Monitor ID）的表格，来标识被监测系统。一个ECU不一定需要支持所有的MID，获知具体支持哪些MID的方法与01和02服务所使用的方法相同，也是先MID，读取00，20，40等ID，06服务的命令格式是SID+若干MID，命令格式如下

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Byte** | **Parameter Name** | **Cvt** | **New Value** |
| **#1** | **Request on-board monitoring test results for specific monitored systems request SID** | **M** | **06** |
| **#2** | **On-board Diagnostic Monitor ID** | **M** | **xx** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Byte** | **Parameter Name** | **Cvt** | **New Value** |
| **#1** | **Request on-board monitoring test results for specific monitored systems request SID** | **M** | **46** |
|  | Data record of supported OBDMID = [ |  |  |
| #2 | On-board Diagnostic Monitor ID | M | xx |
| #3 | Std/Manuf Defined ID#1 | M | xx |
| #4 | Unit And Scaling ID#1 | M | xx |
| #5 | Test Value(High Byte)#1 | M | xx |
| #6 | Test Value(Low Byte)#1 | M | xx |
| #7 | Min Test Limit(High Byte)#1 | M | xx |
| #8 | Min Test Limit(Low Byte)#1 | M | xx |
| #9 | Max Test Limit(High Byte)#1 | M | xx |
| #10 | Max Test Limit(Low Byte)#1 ] | M | xx |
| … | … | … | … |
|  | Data record of supported OBDMID = [ |  |  |
| #n-8 | On-board Diagnostic Monitor ID | C1 | xx |
| #n-7 | Std/Manuf Defined ID#n | C2 | xx |
| #n-6 | Unit And Scaling ID#n | C2 | xx |
| #n-5 | Test Value(High Byte)#n | C2 | xx |
| #n-4 | Test Value(Low Byte)#n | C2 | xx |
| #n-3 | Min Test Limit(High Byte)#n | C2 | xx |
| #n-2 | Min Test Limit(Low Byte)#n | C2 | xx |
| #n-1 | Max Test Limit(High Byte)#n | C2 | xx |
| #n | Max Test Limit(Low Byte)#n ] | C2 | xx |

06服务的Response中，针对某一个MID，可能有多个TID(Test ID),因为针对一个系统可能有多个测试项目。TID表格也在OBD中定义。06服务的response格式固定，每个MID的每个TID有6个部分组成，可以在上面的例子中看出：

1. MID
2. TID
3. Unit And Scaling ID，用于标识这个TID的测试内容是什么，比如电压、时间、计数器之类的。
4. Test Value，实际测量值
5. Min Test Value，这个测量值的最小值
6. Max Test Value，这个测量值的最大值

**Service 07 – Request Emission-Related Diagnostic Trouble Codes Detected During Current or Last Completed Driving Cycle**

07服务也是获取DTC，但是它与03服务区别在于，它用于获取在当前以及上一个驾驶循环中出现的处于“pending”状态的DTC，而03服务获取的是confirmed DTC.

它的请求格式和响应格式如下两幅图：

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Byte** | **Parameter Name** | **Cvt** | **New Value** |
| **#1** | **Request emission-related diagnostic trouble codes detected current or last completed driving cycle request SID** | **M** | **07** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Byte** | **Parameter Name** | **Cvt** | **New Value** |
| **#1** | **Request emission-related diagnostic trouble codes detected current or last completed driving cycle request SID** | **M** | **47** |
| #2 | #cf DTC = [no emission-related DTCs ## of emission-related DTCs] | M | 00  01-FF |
| #3 #4 | DTC#1(High Byte)  DTC#1(Low Byte) | C  C | Xx  xx |
| … |  | … | xx |
| #n-1  #n | DTC#n(High Byte)  DTC#n(Low Byte) | C  C | Xx  xx |

**Service 08 – Request Control of On-Board System,Test or Component**

08服务用于对系统进行控制，进行元件测试操作。它相当于UDS中定义的2F和31服务，它的使用方法是SID+TID,注意这个TID与05和06服务的TID不同，在OBD中有一个专门给08服务使用的TID表格。

**Service 09 – Request Vehicle Information**

09服务用于读取车辆信息，它的命令请求格式是SID+若干InfoType，InfoType在OBD标准中有定义。并不是所有的InfoType都需要被支持，具体哪些InfoType被支持，可以采用和01服务相同的方法来获知，读取00，20，40等ID。比如InfoType = 02代表17个ASCII的Vehicle Identification Number.

目前，UDS和OBD是两套应用层协议，而OBD所提供诊断服务其实属于UDS所提供服务的一个子集，理论上来说UDS中的诊断服务都可以实现OBD中的要求。为了降低同时需要实现两套协议的成本，所以标准化组织分配了ISO-27145（World-Wide Harmonized OBD）这个标准号来将OBD与UDS统一，使用UDS中的诊断服务来替代OBD相关的诊断服务。具体替换方案如下表：

|  |  |  |
| --- | --- | --- |
| UDS | Diagnose Service | OBD |
| 22h | Read Data By Identifier | 01h，09h |
| 19h | Read DTC Information | 02h，03h，07h |
| 14h | Clear Diagnostic Information | 04h |
| 31h | Routine Control | 05h，06h，08h |

比如，在OBD中，使用02，03，07分别读取confirmed DTC，DTC环境数据，pending DTC，而这些功能都可以通过UDS中的19服务来实现（配合上不同的状态掩码和读取参数）。

## OBD服务和UDS服务在DCM模块中如何实现的

DCM模块可以**既支持UDS服务，又支持OBD服务**。我们假设UDS服务和OBD服务底层都是通过CAN总线数据传输，我们需要在DCM配置代码中做如下工作：

创建两个Protocol，分别为UDS和OBD

两个Protocol有不同的优先级，可以设置OBD优先级高于UDS，当产生诊断冲突时，OBD服务优先

建立UDS Service table和OBD Service table

建立OBD的PID list，即支持的PID

当前实现OBD的方案看到有两种：

1. 把OBD服务0x01-0x0A，也归属于UDS服务列表，所以只需要UDS服务一个Protocol，这种并不是正规做法。
2. OBD是完全独立的Protocol，一条消息PDU只能归属于一个Protocol，所以OBD不能复用UDS服务的can报文。

**注：OBD最好是和UDS使用不同的CAN总线。**

**总结**

OBD服务适合尾气排放相关的诊断服务，使用场景不多，一般在燃油车的动力总成ECU需要支持OBD和UDS两个服务。