

UDS 诊断教程（三）

在上一篇文章中我写了 Diagnostic and Communication Management（诊断和通信管理）这一类诊断服务中的 0x10, 0x11, 0x27, 在这篇文章中继续这一大类诊断服务中的其他内容。

CommunicationControl (0x28)

该服务用于打开/关闭某些类别的报文的发送/接收。它通常在刷写软件或大量数据的时候使用，因为在刷软件或参数的时候并不需要 ECU 进行与通信相关的功能，将通信关闭之后可以把所有通信资源都留给软件或参数的下载，当下载过程完成之后再利用该服务将通信恢复即可。

0x28 服务的格式如下图所示

CommunicationControl Request	SID	Sub-function	communicationType	nodeIdentificationNumber
------------------------------	-----	--------------	-------------------	--------------------------

0x28 服务的格式

第一部分即 SID，一个 byte，值为 0x28；

第二部分是 sub-function，表明要对 ECU 的通信进行哪种控制，具体包括：

0x00 enableRxAndTx（激活接收和发送）

0x01 enableRxAndDisableTx（激活接收和关闭发送）

0x02 disableRxAndEnableTx（激活发送和关闭接收）

0x03 disableRxAndTx（关闭接收和发送）

0x04 enableRxAndDisableTxWithEnhancedAddressInformation（激活接收和关闭发送，针对特定的地址）

0x05 enableRxAndTxWithEnhancedAddressInformation（激活接收和发送，针对特定的地址）

0x06 - 0x7F 都是保留或者留给厂商自定义的。

第三部分表明这条诊断请求要对哪种报文进行控制，长度为 1 个 byte，定义如下表所示：

Encoding of bit	Value	Description	Cvt	Mnemonic
0 – 1	0x0	ISOSAEReserved	M	
	0x1	normalCommunicationMessages This value references all application-related communication (inter-application signal exchange between multiple in-vehicle servers).	U	NCM
	0x2	networkManagementCommunicationMessages This value references all network management related communication.	U	NWMCM
	0x3	networkManagementCommunicationMessages and normalCommunicationMessages This value references all network management and application-related communication.	U	NWMCM-NCM
2 – 3	0x0 – 0x3	ISOSAEReserved	M	ISOSAERESRVD
4 – 7	0x0	Disable / Enable specified communicationType See encoding of bit 0-1. In the receiving node including communication to all connected networks. This only disables the node's communication into the connected networks but not the communication of other nodes on the networks (i.e., receiving node is not responsible to disable communication in each node of the network).	U	DISENSCT
	0x1 – 0xE	Disable / Enable specific subnet identified by subnet number	U	DISENSSIVSN
	0xF	Disable/Enable network which request is received on (Receiving node (server))	U	DENWRIRO

communicationType 的定义

这个 byte 中最常用的就是低 2 bit，0x1 代表普通应用报文，0x2 代表网络管理报文，0x3 代表普通应用报文和网络管理报文。

第四部分是 optional 的，只有当 sub-functional 等于 0x04 或 0x05 时才需要使用。

举个完整的诊断服务例子：

28 01 01 表示**激活应用报文的接收**并**关闭应用报文的发送**（网络管理报文不受影响）。

28 00 01 表示**激活应用报文的接收和发送**（网络管理报文不受影响）。

TesterPresent (0x3E)

这个诊断服务的用处可以通过它的名字很明显地得知，即告知 ECU 诊断仪还在连接着。

在上一篇文章中我说到了关于 session 的部分，如果没有诊断命令的发送和接收，ECU 将从 non-default session 中回退到 default session，0x3E 就是用于使 ECU 保持在当前 session。

这应该是 UDS 中最简单的一个诊断服务了，它永远只有两个 byte，格式如下：

A_Data byte	Parameter Name	Cvt	Byte Value	Mnemonic
#1	TesterPresent Request SID	M	0x3E	TP
#2	sub-function = [zeroSubFunction]	M	0x00 / 0x80	LEV_ZSUBF

0x3E 诊断服务的格式

当 sub-function 是 0x00 时，ECU 要给出 response；当 sub-function 是 0x80 时，ECU 不需要要给出 response。

一般来说主机厂会为此服务定义两个时间参数，一个参数用于规定自己的诊断仪发送 0x3E 服务的间隔，另一个参数用于定义 ECU 收不到 0x3E 服务的 timeout 时间。

ControlDTCSetting (0x85)

该服务用于控制 ECU 的 DTC 存储，这个服务常常和前面提到的 28 服务一起使用，比如，在开始写参数之前，为了获得更快的传输速度，我们用 28 服务把所有 ECU 的通信关闭了，但此时因为收到不到相关的报文，ECU 会没有必要地存储很多 DTC，这时如果我们使用 85 服务把 ECU 存储 DTC 的功能暂时性地禁止掉，则不会造成这种麻烦。

ControlDTCSetting Request SID	Sub-function	DTCSettingControlOptionRecord
-------------------------------	--------------	-------------------------------

0x85 服务的格式

第一部分即 SID，一个 byte，值为 0x85；

第二部分是 sub-function，表明是打开还是关闭 ECU 的 DTC 存储，具体包括：

0x01 on

0x02 off

第三部分是 optional 的，由各家自己定义，比如，可以用 FF FF FF 来表示这条诊断命令针对所有的 DTC。

ResponseOnEvent (0x86)

我在以前的文章里说，诊断通信过程是问答式的，诊断仪发请求，ECU 给响应。0x86 服务算是一个例外，在 ECU 收到这条 0x86 服务之后，当 DTC 产生时，它会自动地上报 DTC 及相关环境数据，直到用另一条 0x86 服务来关闭 ECU 的这个行为。

该功能主要用于 ECU 的前期开发阶段，在售后和生产中是不会用到的，而且该服务的格式复杂（即可变的参数很多），执行它还分为好几个步骤，我就不详细写了。

LinkControl (0x87)

这个服务用于转化 ECU 数据链路层和物理层的状态，比如，在高速 CAN 上的 ECU 正常通信速率是 500 kbit/s，但它同时也支持 1M bit/s 的波特率，如果需要刷写大量数据，便可以利用这条诊断服务让 ECU 以 1M bit/s 的波特率进行通信。

这个诊断服务的执行分为两个步骤：

1. 验证 ECU 是否支持要调整到的目标波特率
2. 让 ECU 的数据链路层和物理层转到目标波特率的通信状态

只有当第一个步骤验证通过了，第二个步骤才可以成功执行。