UDS 诊断教程（七）

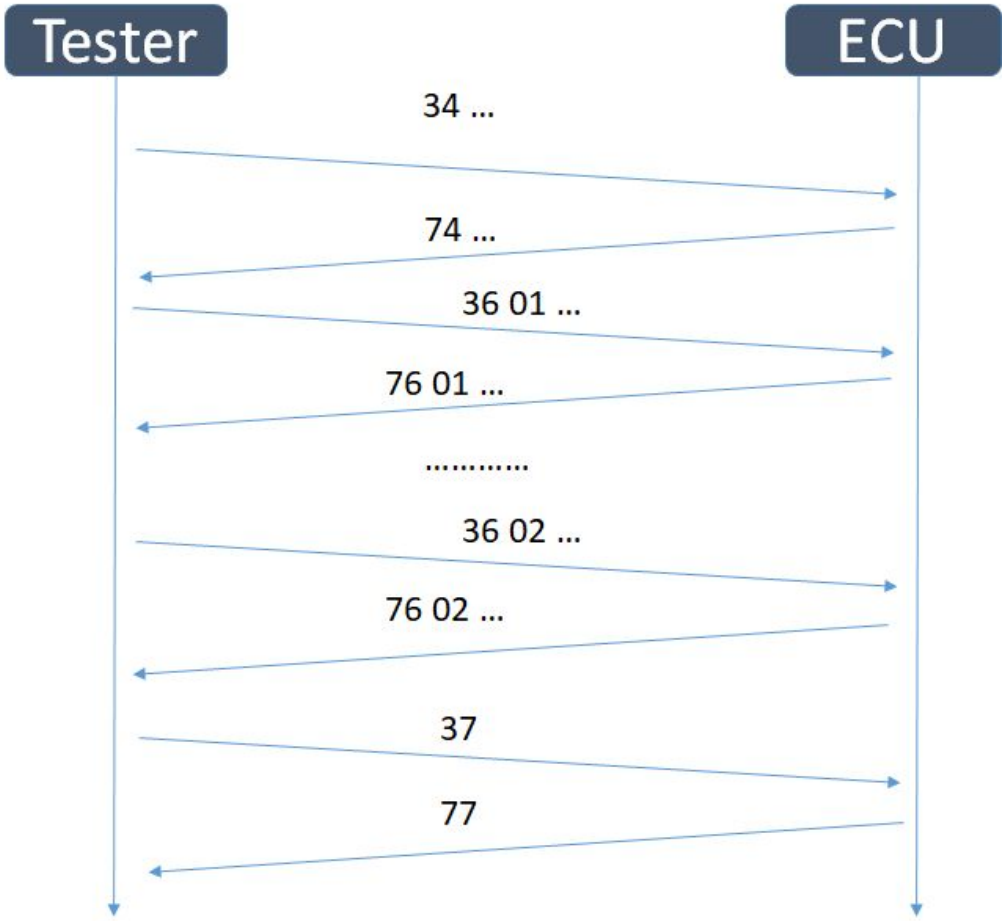
在关于 UDS 的第二篇文章中，我提到过 UDS 定义的服务从逻辑上分为 6 类，在第二至第六篇中已经讲解了前五类，在本文中将介绍最后一类 UDS 服务，即 Upload Download functional unit ，数据的上传下载。

从成本等角度考虑，汽车 ECU 中用于缓存诊断服务数据的 buffer 大小有限，所以当我们需要读取或写入超过 buffer 大小的数据时，就无法简单地使用 2E 和 22 服务了，UDS 据此定义了几个将大块数据写入或读出的服务，即数据下载和上传。

Upload Download functional unit 总共定义了 5 个诊断服务，分别是：

1. RequestDownload （0x34）：客户端向服务器请求下载数据
2. RequestUpload （0x35）客户端向服务器请求上传数据
3. TransferData（0x36） 客户端向服务器传数据（下载），或者服务器向客户端传数据（上传）
4. RequestTransferExit（0x37）数据传输完成，请求退出
5. RequestFileTransfer（0x38） 传输文件的操作，可以用于替代上传下载的服务。

下图是数据下载的简略过程，用到了 34，36，37 这三个服务，如果是上传的话，34 服务被 35 服务替换，数据传输方向变一下，就可以了。



Tester 向 ECU 刷写数据的大概过程

RequestDownload （0x34）：

0x34 服务用于启动下载传输，作用是告知 ECU 准备接受数据，ECU 则通过 0x74 response 告诉诊断仪自己是否允许传输，以及自己的接受能力是多大。

0x34 服务的请求格式

0x34 服务的请求格式包括 5 个部分

第一部分：1 个 byte 的 SID

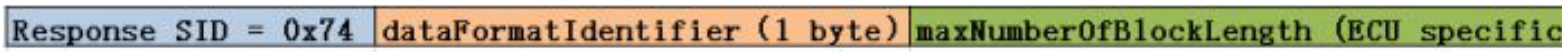
第二部分：1 个 byte 的 dataFormatIdentifier，这里面标识了数据格式相关的信息，比如数据是否有压缩，是否有加密，用的什么算法加密等，应该由主机厂与供应商约定好，用哪个 bit 来表示压缩、加密等信息。

第三部分：1 个字节的 addressAndLengthFormatIdentifier，用于指示后面两个部分所占用的字节，高 4bit 表示 memorySize 所占的字节长度，低 4bit 表示 memoryAddress 所占的字节长度。在这个例子中我将这两个值分别设置为 n 和 m。

第四部分：m 个字节的 memoryAddress，由 addressAndLengthFormatIdentifier 中的低 4bit 指示。含义是要写入数据在 ECU 中的逻辑地址。

第五部分：n 个字节的 memorySize，由 addressAndLengthFormatIdentifier 中的高 4bit 指示。含义是要写入数据的字节数。

ECU 收到请求之后，如果允许传输的话，会给出如下 response



0x34 服务的响应格式

第一部分：1 个 byte 的 Response SID

第二部分：1 个 byte 的 dataFormatIdentifier 作为 echo

第三部分：maxNumberOfBlockLength，长度不定，表示可以通过 0x36 服务一次传输的最大数据量。

TransferData（0x36）：

如果 34 服务得到了正确响应，tester 就要启动数据传输过程了，使用的就是 36 服务。36 服务的格式如下。

0x36 服务的请求格式

第一部分：1 个 byte 的 SID

第二部分：1 个 byte 的 blockSequenceCounter，标识当前传输的是第几个数据块，或者简单地说就是第几次调用 36 服务。

第三部分：transferRequestParameterRecord，传输的数据。第次传输数据量的上限就是 34 服务响应中的 maxNumberOfBlockLength。

举例：如果 ECU 告知 tester,maxNumberOfBlockLength = 20，也就是说 tester 每次通过 36 服务只能发送最多 20 个字节，其中还包括了 SID 和 blockSequenceCounter，所以实际上每次可传的数据信息只有 18 个字节。如果 tester 要传的数据为 50 个字节，则需要传输三次，每次分别传输 18，18，14 个字节，即调用 3 次 36 服务。

36 的响应很简单，就是一个字节的 Response SID 再加一个字节的 blockSequenceCounter 作为 echo。

RequestTransferExit（0x37）：

37 服务用于退出上传下载，如果之前的 34 和 36 服务都顺利执行完成，那么 37 服务就可以得到 ECU 的 positive response。

格式很简单，请求就是 37，正确响应就是 77，都是一个字节。
如果前面的 36 服务没有执行完成，以我前面举的例子来说，比如这个数据块有 50 个字节，但是 tester 只发了两次 36 服务传了 36 个字节，那么这次传输对于 ECU 来说是失败的，所以 ECU 应该给出 NRC 0x7F 37 24，表示诊断序列执行有错误。

关于 UDS 所定义的诊断服务到这里就写完了。接下来我会写两篇文章补充一下 UDS 系列，分别介绍一下 DTC 的 8 个状态位的逻辑关系以及向 ECU 刷写数据或软件的完整流程。在此之后我会写几篇文章来讲述 UDS 在 CAN 总线上的实现，即所谓的 UDSonCAN，涉及到 TP 层的分包、流控制、错误识别等内容，还有基于 CAN 实现的 UDS 中涉及到的各种时间参数。