

TL718 使用大众 CAN TP2.0 协议说明

对于大众 CAN TP2.0 协议，由于用 CAN 自定义参数据的方法与 ECU 通讯对编程有一定的难度，TL718 V1.7 版本新增了 G 号协议，以使通过 PC 串口可方便从大众 CAN 诊断口通讯，并且由 TL718 自动建立及保持通讯链路。

大众很多 CAN 诊断的车可能通过 ISO15765 进入诊断，这个是 OBD2（ISO15031-5）的诊断规范，不在这个讨论范围，可用标准的 OBD2 诊断程序诊断，可参考 SCANTOOL 源代码。

CAN TP2.0 是使用 11 位 ID 500KBPS 波特率进行数据通讯的
TP2.0 的应用层的数据是根据大众 KWP2000 定义的，命令定义及数据流格式相同，编程时用统一的数据转换函数。

A.

TP2.0 的链路分析

首先我们从一个仪表系统的 CAN 诊断过程来了解 CAN 链路。这个过程是 VAS5053 在与 AUDI A6L 仪表通讯过程中用 TL718 ATMA**的数据。

仪表系统诊断过程（TL718**到的数据）

发送和接收数据	数据分析
→200 07 C0 00 10 00 03 01 ←207 00 D0 00 03 51 07 01	系统诊断触发请求地址码 0x07 ECU 应答，系统地址码 0x0751，0x0300 为接收地址，高位在后。
→751 A0 0F 8A FF 32 FF ←300 A1 0F 8A FF 4A FF	建立命令进入系统，并设置 T1 /T3。 T1=10*10ms=100ms T3=50*100us=5ms T1=10*10ms=100ms T3=10*1ms=10ms 第一句是建立连接的请求， 第二句是 ECU 建立应答。 这一步是 TP2.0 协议定义要求的时间参数及进入系统的命令格式。
→751 10 00 02 10 89 ←300 B1 ←300 10 00 02 50 89 →751 B1	10 是设备发送的报文控制字，1 代表请求 ECU 数据，0 是计数值，设备每次请求应将此值加 1，在 0-F 间循环。请求连接诊断 10 89（KWP2000 定义的数据） 02 是设备发送的数据的长度，02 前面的字节 00 不能变。 B1，开始发送多帧数据，序号为 1 高 4 位 B 代表是确认报文，后面无数据，设备不需要应答，ECU 会继续发送后面的数据。1 是序列号，从 1 开始，以后每次设备请求，ECU 都+1，在 0-F 间循环。 10，ECU 应答报文控制字节，高位半字

	<p>1 代表是数据报文的最后一帧数据,接收后需要设备应答,后面是有效的数据,就是说设备完整接收该报文后,需要在 T3 时间内发送应答报文给 ECU。低位字节 0 是一个 0-F 间的计数据值。</p> <p>如果高位半字是 2,同样代表该报文是有效数据,但设备接收到该报文后,应继续接收后面的报文。</p> <p>这是只有一个帧的情况 第一字节为 10 诊断议已接收,</p> <p>751 B1 设备接收后,响应 B1 (1 是计数值,是最后一帧接收到的报文的计数值加 1,在 0-F 间的值。</p>
→751 11 00 02 1A 9B	<p>读版本信息 指令: 1A 9B</p> <p>11 是设备发送的报文控制字,1 代表请求 ECU 数据,1 是计数值,设备每次请求应将此值加 1,在 0-F 间循环。请求连接诊断 1A 9B (KWP2000 定义的数据)</p> <p>02 是设备发送的数据的长度,02 前面的字节 00 不能变。</p>
<p>←300 B2</p> <p>←300 21</p> <p>00 30 5A 9B 34 46 30</p> <p>←300 22 39 31 30 39 30 30 43</p> <p>←300 23 20 20 30 31 32 30 03</p> <p>←300 24 00 08 5F 07 EA 05 EC</p> <p>←300 25 04 74 4B 4F 4D 42 49</p> <p>←300 26 49 4E 53 54 52 2E 20</p> <p>←300 27 4D 37 33 20 48 31 33</p> <p>←300 18 20</p>	<p>返回多包数据,注意 ECU 前面一条响应已经发回过一个数据包,所以这个计数数据值为 B2,同样后面的流数据从 21 开始,因为前条指令已发了一条。</p> <p>30 0x30=48 是数据的总长度,阴影部分总长</p> <p>最后一帧</p>
→751 B9	诊断议已接收,响应 B9 (9 是计数数据值)
→751 12 00 04 31 B8 00 00	读功能选择判断协议,见协议部分
<p>←300 B3</p> <p>←300 29 00 10 71 B8 01 02 01</p> <p>←300 2A 03 01 04 01 06 01 07</p> <p>←300 1B 01 08 01 0C</p>	10 是有效数据的长度,总计 16 字节
→751 BC	
<p>→751 A3</p> <p>←300 A1 0F 8A FF 4A FF</p> <p>→751 A3</p>	空闲帧命令应答,TP2.0 协议规定的格式

←300 A1 0F 8A FF 4A FF	
.	

注：这里有 2 对计数值是独立的，不要混，1、设备请求计数。→ECU 接收到后确认的确认应答计数，是设备的请求值+1。2、ECU 发送数据的计数→设备接收完成后的应答计数是 ECU 的发送最后一帧数据的记数值加一。

A.

TL718 如何工作与编程

为了方便 PC 串编程，TL718 已以为你建立了 TP2.0 数据链路部分数据的通讯，只要使用方便的 OBD 命令可直接读取，发送数据。

TL718 的诊断过程如下：

>atspg →选择协议 G

OK

>atiia07→设定触发地址 0x07 为仪表

OK

>atsw19→设定链路保持时间 1S

OK

>10 00 02 10 89→发送诊断开始命令（11 位 ID 由 TL718 自动设置）

300 B1 →ECU 应答

300 10 00 02 50 89 →ECU 返回的数据，代表进入诊断成功

***** 在这过程中 TL718 做了什么？ *****

- 1、TL718 发送了触发命令
- 2、接收后根据 ECU 的系统自动设置了发送的 11 位 ID ，及接收 ID
- 3、接着发送了建立连接命令。
- 4、连接建立后发送了 10 00 02 10 89 的诊断开始命令。
- 5、接收完成后，自动应答 ECU 接收完成。

就是下列表格内的数据。

接收回来是 ECU 对 1089 请求的应答数据。

- 6、同时如果总线上在 1S 内没有通讯，TL718 会自动发送链路保持数据，并接收以保持链路

正常。

→200 07 C0 00 10 00 03 01 ←207 00 D0 00 03 51 07 01
→751 A0 0F 8A FF 32 FF ←300 A1 0F 8A FF 4A FF
→751 10 00 02 10 89 ←300 B1 ←300 10 00 02 50 89 →751 B1

下一步你只要跟据需要发送各种请求就可

>**11** 00 02 **1a 9b** →读 ECU 版本信息

300 B2
300 21 00 30 5A 9B 34 46 30
300 22 39 31 30 39 30 30 43
300 23 20 20 30 31 32 30 03
300 24 00 08 5F 08 31 0D EC
300 25 15 7C 4B 4F 4D 42 49
300 26 49 4E 53 54 52 2E 20
300 27 4D 37 33 20 48 31 32
300 18 20

(接收完成后的应答与 TL718 内部自动完成，不用应用程序发送)
应用程序接收到这些数据后应进行重组并处理，取消无用的字节。

>**12** 00 04 31 B8 00 00 →读菜单功能项

300 **B3**
300 **29** 00 10 **71 B8 01 02 01**
300 **2A****03 01 04 01 06 01 07**
300 **1B 01 08 01 0C**

>**13** 00 04 18 00 FF 00 →读故障码

300 **B4**

300 1C 00 0B 58 00 00 00 00

读到 0 个故障码

注意：加粗的字节值的变化规律。

每次发送 OBD 诊断请求命令时要求计数值加一，发送的数据不用包含 CAN 帧内的长度字节，长度字节由 TL718 发送时自动插入。接收时的 CAN 帧内的长度字节 TL718 自动取消。所以你使用时没用 CAN 定义的长度字节。

>A8→退出诊断，这是 TP2.0 定义的结束指令

NO DATA

>ATPC →发送 ATPC 指令关闭协议

OK
