

汽车电子 CAN总线

如何进行汽车 CAN 总线开发？

开发流程是什么？需要有那些硬件准备？要学习哪方面的知识？麻烦大神解答一下！

关注问题 写回答 邀请回答 好评问题 16 添加评论 分享 ...

28 个回答

默认排序

王御 吉林大学 车辆工程工学博士

编辑推荐

846 人赞同了该回答

20200417 更新嵌入式平台推荐，飞思卡尔较贵，换成STM32

=====

@蔡小鸟@踩猫尾巴 分别从不同角度贡献了非常不错的回答，请大家参考！

=====

谢谢。

个人认为朱志强的答案是不正确的。（知乎的@怎么用？）CAN总线和基于CAN总线的诊断完全不是一个概念，CAN总线的硬件也不一定是OBD的硬件。至于“多去4S店问问技师，车厂的人大部分的人都不一定比这些人懂的多”，你让这些技师给我写个能用的ECU CAN接口出来？

以下开始正式答题。

首先，请看这份恒润工程师写的论文。

CAN总线系统测试技术

里面基本描述了CAN总线开发的流程，但主要侧重于CAN总线的测试，不过也能让你有所了解。根据你的描述，假设你的要求是：**能够基于某嵌入式平台（最好该平台在汽车ECU开发中被广泛使用），根据给定的通信协议（一般由通信矩阵描述），编写CAN接口代码，并能通过相关设备进行简单测试。**

推荐的学习步骤为：

1. 了解CAN基本知识

可以通过这个ppt，恒润的东西还是不错的。

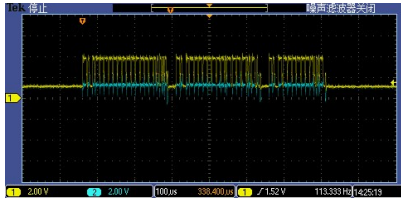
CAN基础_百度文库

2. 直观的了解CAN

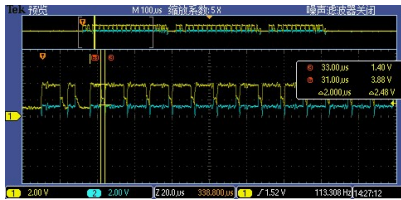
这需要你购买或借用CAN总线检测设备，档次差很多。如果你自己玩玩，可以买个周立功的USB转CAN，淘宝也就几百块钱。如果是课题组采购，可以买个Kvaser的USBCAN，几千块吧。大致这个样子。



要是土豪的话，可以买Vector的VN1600，这个就好几万了。以上三种我都用过。一定要买双通道，然后一个口发，一个口接，可以自己接延长线通过示波器看信号了。尝试修改报文ID，数据，波特率等等，看示波器的变化，对理解第一部分的内容很有帮助。

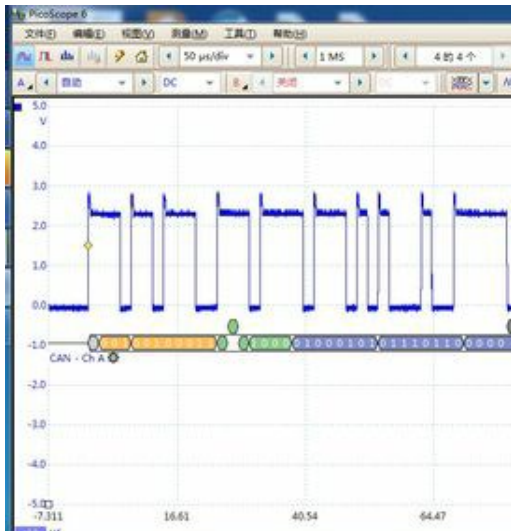


这个图中有三个message。



一位是2us，总线的波特率是多少？ $1/2e-6=500\text{kbps}$

20201224更新一张更好的说明图片



相关推荐



AUTOSAR 规范与车用控制器软件开发
65 人读过

阅读



汽车总线故障检修与典型案例
2 人读过

阅读



汽车车载网络技术详解 第3版
17 人读过

阅读

刘青山 · 知乎指南 · 知乎协议 · 知乎隐私保护指引

应用 · 工作 · 申请开通知乎机构号

侵权举报 · 网上有害信息举报专区

京 ICP 证 110745 号

京 ICP 备 13052560 号 - 1

京公网安备 11010802020088 号

互联网药品信息服务资格证书

(京) - 非经营性 - 2017 - 0067

违法和不良信息举报: 010-82716601

儿童色情信息举报专区

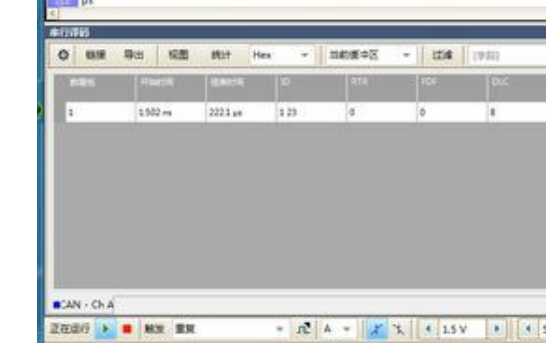
信息安全漏洞反馈专区

内容从业人员违法违规行为举报:

zh-ygwfwgjb@zhihu.com

证照中心 · Investor Relations


联系我们 © 2021 知乎



3. 选择嵌入式平台，学习CAN接口的编程
可以选择STM32F103C8T6（¥16）+TJA1050 CAN 收发器（¥4.5）+ST-LINK下载器（¥18）进行练习，开发软件也是免费的STM32CubeIDE，连上硬件单步调试，很直观。
程序的写法都是差不多的。然后和第二部分的总线检测设备连在一起进行测试，或者买两个STM32F103C8T6+TJA1050 连起来测。可以学习底层寄存器的配置，CAN中断等知识。
应用层需要详细了解信号转换的知识，特别是Big Endian和Little Endian的区别，factor和offset的作用，如何置1或置0，可以参考
DBC_File_Format_Documentation_百度文库
很详细，看完就懂了。
4. 如果有兴趣，可以看看J1939协议（用于商用车）或者CANOpen协议（用于现场控制，很多电机控制器使用）或者ISO 15031-5（用于OBD诊断）。
最后，希望你了解CAN以后，**不要花过多的精力在底层上**，基本都是体力劳动。体现价值的还是上层。请看看一汽技术中心的相关招聘要求：

- 工作职责
- 1.设计新能源汽车网络拓扑和制定网络技术需求；
 - 2.制定新能源汽车网络通信需求、网络通信矩阵和网络数据库；
 - 3.设计和开发新能源汽车网络通信协议仿真试验、网络通信规范、网络线束规范；
 - 4.设计新能源汽车网关控制器及具有相同网络通信功能的接口控制器产品定义与技术规范；
 - 5.负责新能源汽车电子电气开发过程中的问题解决，配合生产和售后服务；

如果对你有帮助，请给我个赞同。
编辑于 2020-12-24
赞同 846 78 条评论 分享 收藏 喜欢 收起

 柏拉涂
你看到的是你想看到的

编辑推荐
等 2 项收录

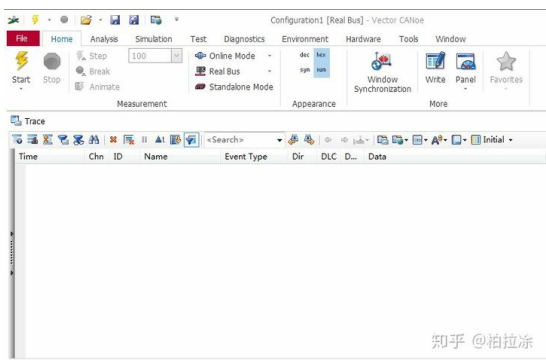
462 人赞同了该回答
1 有汽车电子工程师的地方就有CAN

智能网联汽车的浪潮席卷了全球背景下，大数据、云平台、人工智能、自动驾驶...早已成为汽车电子从业者耳熟能详的词汇。然而，诞生于1986 年的CAN总线技术却是经久不衰，生机勃勃。可以预见的是，在未来很长一段时间内CAN总线还是车载网络的主力军之一。

毫不夸张的说，CAN总线运用是每一位初入汽车电子工程的必修课，无论你是汽车电子硬件工程师、软件工程师，系统工程师，测试工程师，路测工程师，标定工程师.....反正只要你属于汽车电子“搬砖”类工程师，你逃不了带着你的或是示波器、或是CAN OE,Vehicle spy(在车内采集CAN数据，分析各种疑难杂症。没玩过CAN OE,Vehicle spy你都不好意思说自己做过汽车电子开发。



vector VN1640 CAN工具



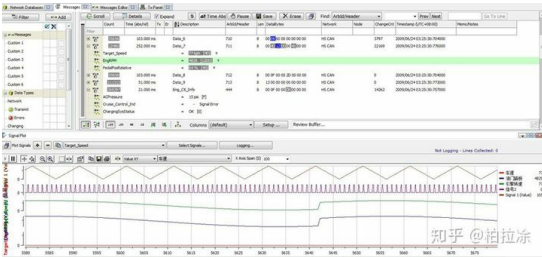
can oe界面





知乎 @柏拉涂

vehicle spy CAN工具



知乎 @柏拉涂

vehicle spy界面

2 CAN基础知识的了解

既然是汽车电子工程师的必修课，就得扎实的学习，经历了数十年的发展与普及网络上关于CAN的材料可以说是数不胜数，大家可以网络检索，这里推荐一本本人学习时看的比较多的一份材料

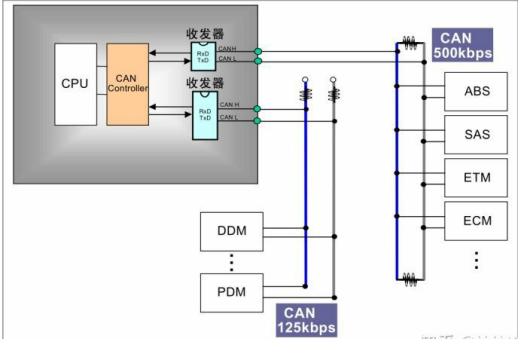
CAN入门书 - 图文 - 百度文库
wenku.baidu.com/view/7701528a6529647d2728520f...

该入门书源自于著名汽车芯片生产商Renesas(瑞萨)，个人感觉该材料特点实战意义很强，比较适合实战开发中学习使用。

这里简单梳理书中的关键字、知识点：

(1) CAN网络

如图，CAN网络通常有N个节点组成，节点间呈总线式连接，每一个节点必须包含CAN收发器、CAN控制器、主控CPU，（通常CAN控制器、主控CPU为集成式方案）。较为常见的CAN通讯速率为500kbps，125kbps，即行业内经常所谓的低速容错CAN和高速CAN，二者具有不同的物理特性。



知乎 @柏拉涂

图3. CAN连接图

CAN连接示意图

低速容错CAN：CAN_H或CAN_L仅有一根断开时，仍可正常通讯，主要使用在对可靠性要求高的场合如车身控制CAN网络通道。

高速CAN：用于更高的数据吞吐能力，主要使用在对实时性、数据传输量大的场合，如汽车动力系统CAN通道等。

高速CAN VS 低速CAN
blog.csdn.net/ppdyhappy/article/details/71498688

(2) CAN帧分类

CAN的帧主要可分为数据帧、遥控帧、错误帧、过载帧、帧间隔。各种帧类型的用途如下表所示，作为数据传递的载体数据帧是最重要的，正常的控制命令、状态信息、诊断数据，刷新数据都是通过数据帧传递的。

表7. 帧的种类及用途

帧	帧用途
数据帧	用于发送单元向接收单元传送数据的帧。
遥控帧	用于接收单元向具有相同ID的发送单元请求数据的帧。
错误帧	用于检测出错误时向其它单元通知错误的帧。
过载帧	用于接收单元通知其尚未做好接收准备的帧。
帧间隔	用于将数据帧及遥控帧与前面的帧分离开来的帧。

知乎 @柏拉涂

(3) CAN数据帧的组成

要理解CAN，一定、绝对、千万不能不知道数据帧的组成，特别是要关注其中的仲裁段，数据段，如下图所示（仲裁段为11bit）的帧格式组成,ID取值范围可为0x000~0x7FF；不同的厂家会将数据段进行区域划分，如（仅做示例）：

0x00~0xFF：用于高优先级的事件性报文传送；

0x100~0x4FF：用于周期型报文的传送；

0x500~0x5FF：用于网络管理报文的传送；

0x600~0x6FF：用于调试开发、标定相关报文的传送；

0x700~0x7FF：用于诊断相关报文的传送；

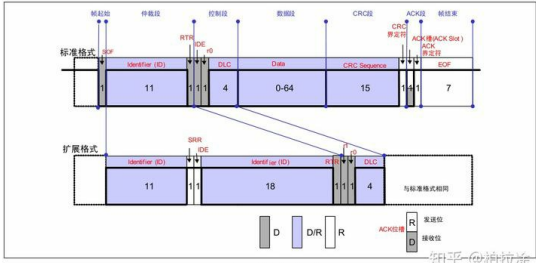
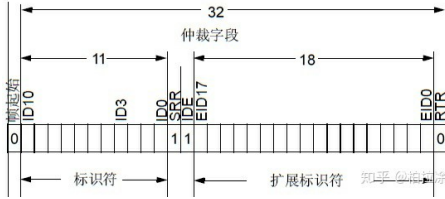


图 11. 数据帧的构成

标准帧的构成

扩展帧：

相对于上图的标准数据帧，还有扩展CAN 数据帧，CAN 数据帧中紧随SOF 位的是32 位的仲裁字段。仲裁字段的前11 位为29 位标识符的最高有效位（Most Significant bit，MSb）（基本ID）。紧随这11 位的是替代远程请求（Substitute Remote Request，SRR）位，定义为隐性状态。SRR位之后是IDE 位，该位为隐性状态时表示这是扩展的CAN 帧。



3 汽车开发中CAN开发的主要内容

读完入门书，对CAN的概念有了一个初步的了解下面简单谈一下CAN开发的方面。

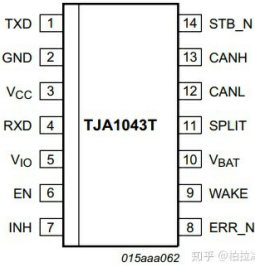
由于本人涉及较多的为软件类开发，对于硬件开发这里不做太多描述。

（1）CAN驱动开发

要实现CAN的收发必须先实现CAN驱动开发，CAN驱动开发主要包括：CAN控制器驱动开发和CAN收发器驱动，较为经典的NXP的TJA104X系列CAN收发器，大部分NXP MCU集成的flexCAN控制器。这部分开发更多的是阅读对应型号的CPU，控制器的芯片手册，结合示例代码，一句一句敲，一行一行敲，再配合示波器（逻辑分析仪），各类CAN工具进行反复摸索。

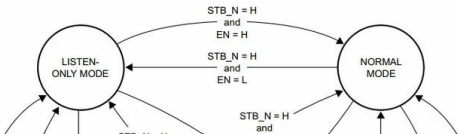
Register
Module Configuration Register (MCR)
Control 1 register (CTRL1)
Free Running Timer register (TIMER)
Rx Mailboxes Global Mask register (RXMGMASK)
Rx Buffer 14 Mask register (RX14MASK)
Rx Buffer 15 Mask register (RX15MASK)
Error Counter Register (ECR)
Error and Status 1 Register (ESR1)
Interrupt Masks 2 register (IMASK2)
Interrupt Masks 1 register (IMASK1)
Interrupt Flags 2 register (IFLAG2)
Interrupt Flags 1 register (IFLAG1)
Control 2 Register (CTRL2)
Error and Status 2 Register (ESR2)
CRC Register (CRCR)
Rx FIFO Global Mask register (RXFGMASK)
Rx FIFO Information Register (RXFIR)
Message buffers
Rx Individual Mask Registers

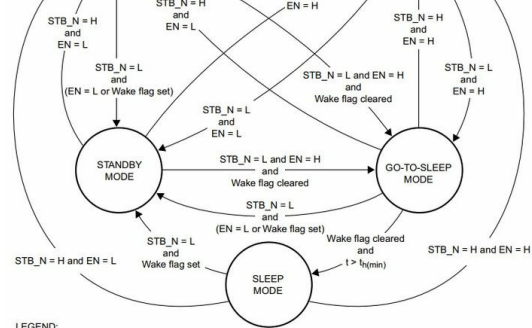
NXP 某芯片FlexCan相关主要寄存器



TJA1043T示意图（来自NXP官网截图）

CAN收发器驱动的开发中最最关键的一部分工作就是了解收发器不同工作模式的切换方法，如下图TJA1043T收发器状态切换示意图，收发器工作时主要分为NORMAL Mode,STADBY Mode,GO-TO-SLEEP Mode,SLEEP Mode。





LEGEND:
= H, = L
logical state of pin

Fig 4. Mode transitions when valid V_{CC}, V_{IO} and V_{BAT} voltages are present

TJA1043T收发状态切换示意图（来自NXP官网截图）

（2）CAN通讯矩阵

CAN通讯矩阵通常由整车厂完成定义，车辆网络中的各个节点需要遵循该通讯矩阵才能完成信息的交互和共享。

如图为vector工具打开XXX.dbc文件（常用的保存通信矩阵文件格式）后的示例，可以看到CAN报文Message1单次可传送8bytes，即64bits信息，64bits由多个signal组成，各个signal分布在message的不同位置，（示例）其中蓝色的openwindow可表示为车窗打开控制指令。

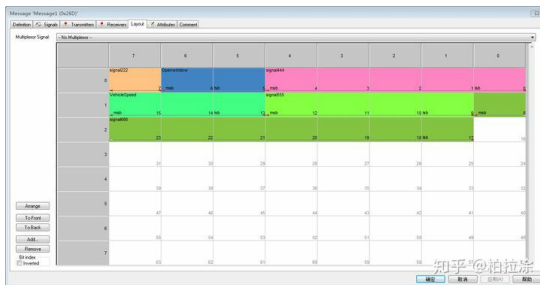
如：

①openwindow=0时，表示打开车窗，openwindow=1时，表示关闭车窗；

②vehiclespeed表示车速信息，vehiclespeed=5表示5km/h。

这里只做简单示例，实际汽车开发中还会涉及到一定的物理值与逻辑值的转换。

这样当Message1发送到CAN总线上时，接收到CAN节点的就能获取到此时的CAN控制指令或状态值。



dbc文件中message1的示意图

值得注意的是在定义通讯矩阵时，不同的厂家可能会有不一样的选择，如：采用摩托罗拉格式还是英特尔格式不同的厂家的抉择就各有所爱。

Intel格式与Motorola格式的区别

blog.csdn.net/bingdianlanxin/article/details/41522373

（3）基于CAN的车辆诊断

百科——汽车诊断：在不解体（或仅卸下个别零件）的条件下，确定汽车技术状况，查明故障部位及原因的检查。包括汽车发动机的检测与诊断，汽车底盘的检测与诊断，汽车车身及附件的检测与诊断以及汽车排气污染物与噪声的检测等内容。CAN就能很好的满足上述要求。

汽车诊断的开发是汽车电子电器开发中非常重要的一环，对于CAN诊断最为常见的是UDS。UDS协议即ISO14229,是Unified Diagnostic Services，统一诊断服务，是诊断服务的规范化标准，在汽车诊断方面广泛使用，如图，为满足诊断需求，UDS中定义了一系列的服务。

SID	描述
0x10	DiagnosticSessionControl
0x11	ECUReset
0x14	ClearDiagnosticInformation
0x19	ReadDTCInformation
0x22	ReadDataByIdentifier
0x27	SecurityAccess
0x28	CommunicationControl service
0x2E	WriteDataByIdentifier
0x31	RoutineControl
0x3E	TesterPresent
0x85	ControlDTCSetting

当然，为了确保诊断报文的稳定传输，还有ISO 15765协议是一种CAN总线上的诊断协议。

其中：

ISO 15765-1包括物理层和数据链路层，

ISO 15765-2对网络层进行说明，

ISO 15765-3则是规定到应用层的具体服务。

上述标准内容较为复杂，此处不做展开，决心要做好CAN开发的同学，尤其是CAN诊断的同学，熟读ISO14229,ISO 15765是不可避免的了。

OSI	Standard
Application (Layer 7)	ISO 14229-1/5
Presentation (Layer 6)	OEM specfc (J1939,

etc.)	
Session (Layer 5)	ISO 14229-2
Transport (Layer 4)	ISO 15765-2
Network (Layer 3)	ISO 15765-2
Data Link (Layer 2)	ISO 11898 (CAN)
Physical (Layer 1)	ISO 11898 (CAN)

基于CAN诊断的协议示意图

(4)基于CAN的刷新

由于设计缺陷或者功能升级，车载控制器在生命周期内会有软件刷新的需求，作为控制器与外界几乎唯一的数据通道，车载控制器的软件刷新通常由CAN通道实现，基于CAN的刷新又与基于CAN的诊断息息相关。

(5) CAN网络管理

以后有机会再细聊，主要可分为OSEK网络管理，AUTOSAR网络管理。

4 进阶阶段-不得不提的AUTOSAR

AUTOSAR在汽车电子行业的知名度应该不会亚于“六神”在中国香水界的地位。简单的讲AUTOSAR是由全球汽车制造商（ 宝马、戴姆勒、福特... ）、部件供应商及其他电子（ 大陆、博世... ）、半导体和软件系统公司联合建立，各成员保持开发合作伙伴关系。自2003年起，各伙伴公司携手合作，致力于为汽车工业开发一个开放的、标准化的软件架构。

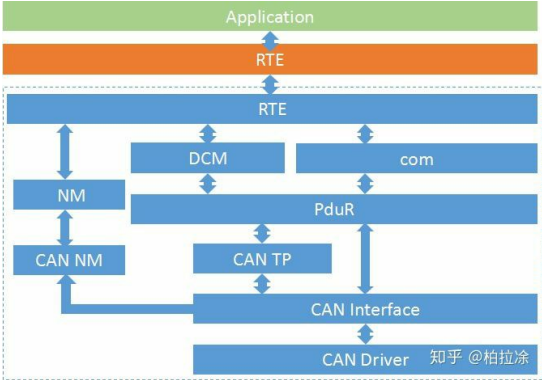


AUTOSAR联盟创始成员

CAN作为汽车电子领域最为重要的通讯形式，AUTOSAR怎么可以不对CAN进行定义规范，可以说AUTOSAR的架构思想对现今的软件架构产生了重要影响，如图为初步整理的的AUTOSAR中关于CAN的相关模块及架构，可以清楚的看到其中由下至上系统的定义出了CAN驱动、接口层、传输层、 CAN诊断、 CAN网络管理等。

目前国内各大整车厂，零部件供应商也在积极推进AUTOSAR，所以要成为CAN进阶高手，绝对少不了好好认识AUTOSAR。

----不过AUTOSAR实在是博大精深，以后有机会再一起深入学习，这里只抛砖引玉。



5 实干，实干，实干

《XX技术从入门到精通》，《手把手教你玩XXX》，《十天精通XXX》，相信这一类的书籍大家都早就耳闻，有幸者可能也拜读过一二，但是到底读完书能不能真的就精通了，或者玩转了？我相信光靠一本书的得道者微乎其微，毕竟不是每个人都是张无忌，随随便便捡到一本乾坤大挪移就能在数个时辰内练成神功。

实践出真知，实践是检验真理的唯一标准这些都是亘古不变的道理。要学好CAN我相信你少不了：

- （ 1 ）一套CAN开发版，淘宝里该类型的开发版比比皆是，售价几十到几百不等；
- （ 2 ）一个CAN数据读写工具，也能轻而易举的花个几百元够到，当然有项目条件或者家里有矿系列可以直接上vector的VN1640，英特佩兹的vehicle spy；
- （ 3 ）逻辑分析仪（ 可选 ），不想把对于CAN帧的了解仅仅停留在纸面上，相对CAN进行更为底层了解的同学可以备一个逻辑分析仪。

然后就是在实践中检验真理，在实践中总结经验。

总结：

我个人觉得吧，对于CAN的学习可能分为以下几个层次（ 阶段 ）：

★

了解、使用CAN：读一读CAN入门书，学习一下CAN工具，基本上你就能胜任CAN报文采集，刷新（使用）；

★★

进一步掌握CAN的工作原理：在1的基础上搞一套开发版，读一读芯片手册，敲一敲代码，示波器测一测波形；

★★★

从事汽车CAN开发：在上述基础上，熟读各类CAN标准（ISO14229，ISO11898，ISO15765），系统学习CAN驱动，CAN诊断、CAN网络管理等知识，并在实战中不断提升认知，积累经验。

★★★★★

成为CAN大神：对AUTOSAR架构融会贯通，熟悉了解autosar中定义的各个CAN模块的功能，工作原理，实现方法。

题外话：CAN作为一项发展普及了数十年的技术，技术标准、工具练已经相当完善，虽然说CAN在未来很长一段时间内还将继续存在，但难免潜力不足；再说汽车网络技术也不乏众多后起之秀虎视眈眈，其中车载以太网可谓是最有潜力者，可多关注。


关于车载以太网推荐本人另一篇回答：

[有量产汽车不用CAN总线而用以太网吗？](#)

码字不易，欢迎赐赞，更多信息请点关注。

编辑于 2019-04-14

[赞同 462](#) [29 条评论](#) [分享](#) [收藏](#) [喜欢](#) [收起](#)

 踩猫尾巴
汽车电子软件攻城狮

编辑推荐

164 人赞同了该回答

/* 没想到随手写的一个回答上了推荐，等有时间我会完善一下答案 */

/*

Revisions:

20161007, 初次提交；

20161011, 增加对第一句话的解释；

20161012, 调整解释的位置，叙述调整；

20161031, 增加回复回复的回复。

*/

我看了下所有的回答，看来国内熟悉CAN总线协议栈研发的工程师比较少。这可能和国内几乎没有Vector、EB、Mentor Graphics这样的第三方供应商有关，产业链上其实缺少了一小环Tier 2。

王博士 @Wang Yu说的内容没什么问题，但看起来有点学院派，所以我想以从业者的视角做一些补充。

已经有主机厂同行大致介绍了主机厂网络组平时与CAN总线相关的工作，因此这里仅介绍OEM以下的内容：

事实上王博士提到的“底层”，在汽车行业的Tier 1供应商中，是非常重要的这一块儿，大多知名零部件供应商都会有专门的团队负责这部分模块的研发，比如Bosch为此收购了ETAS的团队（也因此ETAS现在虽然仍然在售OS，但其中的核心代码是不出售的）。这个“底层”在Tier 1中一般会称做平台，如Visteon正在服役中的有Newton 2.0、Kepler 1.0等平台。同时也会有另一个团队进行下一代平台的研发，和T圈的产品迭代其实挺像的。

补充一下，王博士的观点也没有错，对于应用工程师来说，底层本身是不需要投入过多关注的，就好像程序员一般不需要了解编译器开发团队的工作。（当然这个类比不够准确，与现在行业的状态其实不太匹配。）

而CAN协议栈，就属于上述（软件）平台中的一部分。CAN总线从驱动层往上，有一系列的软件模块，如果想了解最新的分层设计，可以参考AutoSAR中通信部分的文档。而王博士回答中所提到的工作只包含了驱动开发的前期内容，也就是使CAN总线“通”起来，但这个工作在CAN总线的软件开发中其实只是基础的准备工作，一个懂单片机的应届生可能也只需要一两周，就能实现一个能通信的CAN驱动，然而这个驱动在正式产品中是无法使用的。

在实际的商业化产品中，CAN驱动层也会包含一些复杂的策略，如各种异常的处理，以及性能优化。模块本身还会有可配置性和可移植性的要求，毕竟CAN驱动也是软件平台的一部分，可复用性是一个很重要的考量。

而从架构角度来说，在传统的OSEK/VDX架构下，CAN驱动以上就有交互层、传输层（或叫网络层）、诊断协议层、网络管理、标定、信号/报文/协议网关等上层基础软件模块，几乎每一个都有相应的国际标准来对应，是一个相当专业且门槛不低的行业子领域。也因此，除了知名的Tier 1公司会自己招募一个团队做软件平台以外，一些中小型公司往往会选择购买市场上的第三方基础软件，以降低研发成本。目前国际上知名的第三方供应商有上述提到过的Vector、EB、Mentor Graphics、ETAS、KPIT等，国内现在也有普华、恒润等公司在做此类业务。

对于车载ECU来说，Tier 1的研发大致可以分为系统、硬件/电子、软件、机械/结构几个条线。CAN总线涉及最多的是上文提到的软件条线。系统部分基本只需要与OEM的网络工程师及DRE做一个对接，定期沟通下信号需求、通信矩阵释放、样件提交、测试结果反馈等，真正涉及到技术的工作比较少。硬件部分基本都有现成的方案，主机厂的CAN节点企业标准中往往也会提供参考电路和元器件型号推荐，对于有经验的硬件工程师，通常不会出现设计电路有太多的物理层问题需要修改的情况，偶尔可能会有一些电压、斜率之类的小问题。结构条线一般不会有专门与CAN总线相关的工作。

以上内容展开则太过繁杂，不是一篇回答能够全部涵盖的，因此仅作大致介绍，希望能够有所帮助。

这里得解释一下：第一句话是我之前随手写这个答案的时候，表达了工作到现在一直以来的一个感受。可能会让很多汽车工程师觉得被冒犯，这里先表示道歉。其实我没有冒犯的意思，之前写的太随意了，表达不够清楚。相信看到我这个答案的同仁们，很多都称得上熟悉CAN总线，只是之前我想表达的，是其中一个特定方面的熟悉。

其实我的意思是，我觉得国内汽车电子行业内，对CAN总线相关协议有深刻理解及开发经验的工程师的数量，相对于整体的研发规模，有点少了。

比如说，国内数万汽车电子软件工程师，用过CAN总线的十之六七，但知道CAN相关各个模块设计原理的，恐怕百不足一。

比如说，国内主机厂的网络工程师同仁们，对CAN总线相关的标准如数家珍，日常工作就是指导供应商们实现相关协议，对提交的软件进行测试与反馈。但又有多少人，去研究过这些标准背后的know how呢？拿网络管理举例，每家主机厂都会有自己的一份网络管理企业标准，描述了一个详细的实现机制。但是，网络管理的协议并不少，有国际标准如AutoSAR CanNm，OSEK Direct NM，有第三方供应商设计的机制如Vector NmBasic，有主机厂设计完善的完全可以当作一份标准的机制如GM GMLAN NM，Volvo Volcano NM，也有主机厂设计的只有较为简单的状态机如Nissan/Renault NDS中所定义的，甚至还有隔壁商用车所采用的J1939 NM，等等。这么简单的，复杂的，主从的，多主的，其优缺点是什么？选择的依据是什么？不知道在日常工作以外，会有多少人会去进行调研和分析呢？恐怕绝大多数人，像我一样，只能说出一些泛泛的道理，而缺乏有力的实验和数据支撑，以及深刻的洞悉。

需要特别说明的是，由于术业有专攻，我所在的工程师岗位相对会更加关注协议的原理和实现，但这并没有什么高下之分，只是行业内的分工不同。

说到know how，这里还想夹带一些私货，不喜欢的朋友可以跳过。

记得有次我去国内某自主品牌主机厂搬砖，看到实验室中几位工程师在用测力计测试一款进口车型的车门铰链的阻尼大小（这方面不太懂，不知道专业的说法是什么）。当时我就脑洞大开了一下。众所周知，我们国家的工业技术从起步开始，到现在为止，逆向工程从未停歇。这很正常也很必

更，是我们快速追赶发达国家的重要方法。但有时候我在想，其实我们的积累已经不少，是不是该多做一些正向的研发了呢？逆向工程虽然能让我们得到一个接近完美的策略或数据，但通过逆向，我们永远只能知道what和how，而无法知道why。拿CAN总线举例，很多软件工程师都知道一般当错误计数器累加到96的时候，CAN控制器就会报一个Warning，但有多少人知道这个数字为什么是96呢？

回到车门铰链阻尼的选择，我是外行，假如让我来考虑正向开发，那我可能会想到，这个实验需要做出所有合理阻尼范围铰链，然后招募不同年龄段、不同性别、不同人种、不同体型。甚至不同职业的人前来测试，并记录下他们的感受，形成一个数据详实的调查报告，再加上各种危险工况的考量，来确定一个合适的数值。当然这个想法过于复杂了，可能国外都没有这么理想化，但我想说，只有通过类似的正向研发，我们才不会陷入“宝马用20N，那我们就用20N吧肯定不会错”的研发惰性中（数字仅为举例）。

有些技术标准，国外一更新，国内就懵了，只能紧跟步伐，根本不明白国外为什么这么变更。非常希望这样的情况，能够通过我们的努力越来越少。

逆向只能追赶，只有正向，才有可能超越。

愿与怀有技术情结的工程师同仁们共勉。

@只是学电的 关于CAN总线映射到OSI的分层我不是特别了解，因为没有看过ISO/IEC 7498-1等标准。这里提供一些我个人的看法。

前两天正好看到了一个问题个人计算机里面包含网络7层协议的所有协议吗，每一层分别对应哪个部分？。计算机技术，其中@invalid s的回答讲得特别好。对于实际开发工作来说，了解七层模型几乎没有实质上的意义，而且我个人觉得即使是用于教学，以七层模型来做讲解也是很糟糕的做法，因为框架性的内容对于概念都还没建立起来的学生是非常苍白的，对产生感性认识毫无帮助。我比较倾向于在培训时从底层（物理层）开始讲起，多举一些例子，这样即使毫无基础也不至于听完一头雾水。OSI七层模型的学术价值可能更高于实用价值。

回到CAN总线，其实这个问题ISO是已经提供了参考的，在ISO 14229-1: 2013的Introduction中，有一个表格Table 1 — Example of diagnostic/programming specifications applicable to the OSI layers（其实一系列的标准里都有提到，且内容还不太一样）：Application (layer 7) - ISO 14229-1, ISO 14229-3, further standards
Presentation (layer 6) - vehicle manufacturer specific
Session (layer 5) - ISO 14229-2
Transport (layer 4) & Network (layer 3) - ISO 15765-2
Data link (layer 2) & Physical (layer 1) - ISO 11898-1, ISO 11898-2

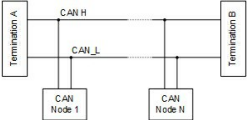
从ISO提供的信息可以看到，在具有诊断功能时，CAN总线对OSI七层模型基本都有映射。而单独考虑通信功能时，在ISO 11898-1: 2003章节7.1 Reference to OSI model中提到，CAN总线只对应了DLL与PHY层。

我个人的理解：OSI七层模型中只有物理层、数据链路层是硬件来实现的（CAN总线线束、CAN控制器、CAN收发器及相关元器件），而其余的五层均为软件实现。

@Keith 很抱歉前段时间比较忙，回复晚了。

我是软件工程师，不了解硬件原理，因此只能告诉你我所知道的，而且不能保证完全正确。。这问题还挺复杂的，我尝试做一下分解。

首先，ISO 11898-2: 2003中给出了CAN总线物理连接方式的建议图：

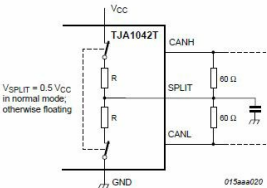


图是我自己画的，原图请参考标准。

可以看到，在预想的状态下，终端电阻与CAN节点（包含CAN硬件的控制器）是分离的，终端电阻处于线束两端的位置（这和终端电阻的原理有关，我不懂本质这里就不妄加说明了），标准也特意强调了这一点。

终端电阻的阻值是一个经验值，标准在7.5.2 Termination resistor章节给出了标称值120欧、最小值100欧和最大值130欧，同时也说明了，根据不同的拓扑、比特率和电压转换速率，偏离标称值是可能的，并且，在每个项目中测试一下其它阻值的适用性也是必要的。

而现在主机厂在设计CAN线束时，一般并不会把终端电阻做在线束上（这也需要主机厂的同仁来解释具体原因，推测和下文提到的电气特性或架构设计有关）。目前一般将终端电阻设计在电控单元的PCB上，参考电路如下：



图源自TJA1042 Product data sheet Rev.9 - 23 May 2016，飞思卡尔（划掉）NXP（划掉）以后可能是高通的官网可下（截止至2016.10.31还是NXP）。

这样分裂终端电阻设计的好处是可以通过一个参考电平稳定电压，降低共模干扰。

主机厂通常会在CAN节点企业标准中推荐这样的参考电路，一条总线上CAN节点的个数一般大于两个，而CAN总线又只需要两个终端电阻，因此就引入了终端节点和非终端节点的概念。

终端节点一般设计在线束的最远端，近似标准中CAN总线的物理连接方式，其它节点则设计为非终端节点。由此可知，终端节点上的终端电阻应为120欧，若为分裂式设计，则应为两个60欧电阻。而非终端节点通常使用一个较大的阻值（具体原理不了解，不知道是要模拟开路还是其它原因，但由于电阻是并联的，至少要保证加上非终端电阻后，总线间的等效阻值与不加之前相比近似相等），如1.3千欧 * 2。

因为我不怎么关心元器件BOM，目前唯一有印象的就是有项目采用过60欧 * 2和1.3千欧 * 2的典型值（500kbps的波特率）。至于9千多欧我没有见过，如果确实无误，那应该是主机厂出于电气特性需要所计算出的非终端节点的阻值。

@chasechoose 不知道你所说的USB转CAN是什么工具？这个问题有点笼统，我不知道该如何说明，如果能描述得详细一点就好了。

每个工具的情况可能有所不同。以我以前开发过的一款WIFI - CAN仿真与测试工具来举例，设备本身是可以获取到报文发送结果的。我们打开了CAN控制器的自发自收功能，然后做了一个回读超时机制。（当然这个问题其实很复杂，还需要确认收到的报文确实是自身发送的而不是其它节点所发送的相同CAN ID的报文，这里不作展开）。当检测到回读超时，就可以判定发送失败了。

因此，如果工具在做二次开发的接口时考虑到了检测发送结果的需求，是可以增加一个发送状态查询接口的。

如果设备本身不提供此类接口，你也可以自己做一个回读判断，来确定发送是否成功。

如果你的疑问是CAN总线怎么保证通信可靠性，那么就发送结果判断这一点而言，所有CAN控制器都提供发送slot状态查询寄存器，CAN驱动可以利用这一特性来实现相应的检测功能。


但是在实际使用中，对于应用报文是可以不做此类查询和通知功能的。

在应用层，这个问题可以由接收端的超时监控来覆盖，当超过一段时间无法收到对应的报文时，接收端会记录报文超时故障乃至节点丢失故障，进而执行异常处理。

在应用层以下，从较底层的CAN总线Error Detection、Error Signalling和Fault Confinement机制到较上层的网络管理机制都有可能覆盖到这个问题。

总体来说，CAN总线功能的实现较复杂，检测机制的冗余性较多，是一个可靠性足够满足大多数车辆信号交互需求的总线。

编辑于 2016-10-31

 尚德机构

广告

在无锡，不用辞职就能读985研究生！双证到手，只考2门！

知名院校热门专业只考2门，学信网终身可查，300分满分170分就能上，点击咨询985/211院校专业信息与学费，获得一手信息，快速上岸 [查看详情](#)

 bingbing 
长安大学 车辆工程硕士

122 人赞同了该回答

结合工作经历和大家分享汽车行业Tire 1供应商是如何进行CAN总线开发的：

开发流程：

ASPICE

所需硬件：

集成了CAN控制器的MCU

专业软件：

CANoe、CANalyzer、Vehicle Spy 3

编程语言：

C/C++、Python、CAPL

相关知识：

CAN/CAN FD协议

UDS诊断协议

Autosar

1.开发流程

ASPICE全称是Automotive Software Process Improvement and Capability Determination，于2005年由AutoSIG发布，是SPICE(ISO\IEC15504国际标准)在汽车行业的衍生标准，其关注汽车行业的软件过程改进和能力测定。ASPICE兴起于欧洲，广泛用于主机厂以及供应商企业自身的的过程能力改进，以及对供应商的风险评估。

CAN总线开发作为车载ECU软件开发中的重要一环也需要遵循ASPICE开发流程。

2.所需硬件

车载ECU通常是选用恩智浦、瑞萨、英飞凌等国际著名半导体芯片供应商提供的车规级芯片，这些芯片一般都集成了CAN通信所需的CAN控制器。芯片的用户手册和数据手册会详细描述CAN控制器的相关资源和设置流程，芯片供应商一般也会提供对应的Demo程序帮助开发者完成CAN驱动的开发。

MCP2515 / STM32 F10x / RH850 芯片中英文手册：

百度网盘链接：pan.baidu.com/s/1Z10-5n ...

提取码：akx5

初学者可以够买一块STM32开发板来学习CAN通信，某宝上正点原子和野火的开发板挺不错的，都有配套的视频和例程，适合入门学习。

视频教程：







商品链接：



野火秉火STM32开发板 ARM开发板51单片机 M3F103高配置

淘宝

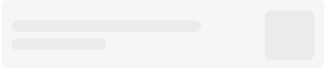
¥ 346.80

去购买 >

在树莓派上使用Python控制MCP2515模块进行CAN通信也是蛮有趣的，可以参考我这篇文章：



商品链接：



3.专业软件

商业项目CAN通信和诊断的开发需要用到专业的软件进行仿真、测试和分析。这些软件工具提供了很多方便实用的功能帮助开发者完成ASPICE开发流程所需的各种软件测试。

相关软件：

CANoe：车载网络仿真、测试、分析

Vehicle Spy 3：车载网络仿真、测试、分析

Diva：诊断自动化测试

VectorCAST：自动化软件测试

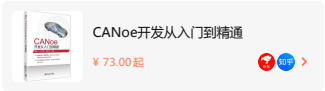
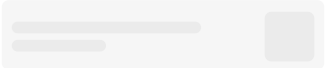
.....

这其中非常常用，功能也非常强大的软件是CANoe。CANoe需要结合Vector提供的硬件板卡一起使用，这些板卡一般都十分昂贵，带有license的低端板卡如VN1630也需要数万元。好在Vector提供了Demo版CANoe软件，强烈推荐初学者下载学习。

Demo版软件地址：



CANoe教程：



4.编程语言

C/C++

C和C++是嵌入式领域应用最广泛的语言，CAN模块的底层驱动以及通信诊断功能的开发大都是基于C/C++，所以学好C/C++是进行CAN总线开发的基础。

Python

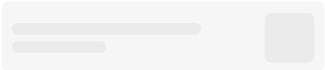
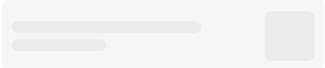
Python在数据处理和自动化测试领域应用十分广泛。利用好Python可以大大提高CAN总线的开发效率。下面介绍两个具体应用场景：

- can matrix 和 数据库 相互转化

CAN 数据库是OEM和电子零部件供应商开发和测试阶段必须用到的文件，通常由OEM提供。CAN数据库一般是可以导入CANoe的.bdc或.arxml格式的文件，将数据库导入CANoe就可以很方便的对总线环境进行仿真、测试和分析。

但有时候OEM不会直接提供.bdc或.arxml文件，而是提供记录总线消息的excel格式的can matrix文件。为了开发和测试需要供应商就必须根据can matrix来编辑生成可导入CANoe的.bdc或.arxml文件。

Vector提供了CANDb++这个软件来帮助用户手动编辑.bdc文件，但根据can matrix定义的消息和信号逐个手动编辑耗时、耗力且容易出错。如果我们熟悉.bdc文件的规则就可以写一个Python脚本实现can matrix和.bdc文件的相互转化。



- 批量生成代码

CAN通信和诊断涉及到很多信号和服务，针对这些信号和服务的软件代码和测试case大都是一样的，我们可以使用Python解析数据库中的信号来生成软件代码和自动化测试case。

CAPL

CAPL全称Communication Access Programming Language，是Vector为CANoe开发环境设计的类C语言。

CAPL应用场景

- 节点仿真

关联Simulation Setup中的ECU节点，实现ECU节点仿真和整车网络仿真。

- 测试功能

关联Simulation Setup中的Test Module模块或是Test菜单中的Test Setup，结合TSL(Test Service Library，测试服务库)进行测试功能开发。

- 分析功能

关联Measurement Setup窗口功能模块中的Program Node，实现总线过滤、分析功能。

5.相关知识

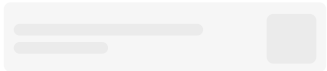
CAN在实际开发中有两个应用方向，通信和诊断。

Vector的入门教程非常不错，推荐给大家：

CAN/CAN FD协议



UDS诊断基础介绍



AUTOSAR

AUTOSAR全称汽车开放系统架构（Automotive Open System Architecture），由全球汽车制造商、部件供应商及其他电子、半导体和软件系统公司联合建立，致力于汽车工业开发一个开放的、标准化的软件框架。

CAN Stack在AUTOSAR架构中的位置

目前主流的汽车零部件供应商大都基于AUTOSAR开发车载ECU产品。AUTOSAR展开来说话就长了，本文就不做过多介绍。

AUTOSAR详细介绍请参考官方文档：



先写到这里，想到再更，谢谢关注，谢谢~

编辑于 2020-11-09

赞同 122 15 条评论 分享 收藏 喜欢 收起

Momenta 已认证的官方帐号

35 人赞同了该回答

汽车诊断协议的发展

当今车辆的电控系统越来越多，比如发动机电控系统ABS、SRS、电动车窗、悬挂、电动箱子等等。与此同时，监控已经遍布于车身的各种传感器，我们能够获取实时状态信息，并将这些信息发送到相应的控制单元当中。每个控制单元同时可以接受各种各样的信息，并对各种信息进行处理、分析，随后发出相关的指令。比如发动机电控系统可以接受来自进气压力传感器、发动机温度传感器、油门踏板位置传感器、发动机转速传感器等信息，经过分析和处理之后会发送相应的指令来控制燃油喷射的喷油量，比如说燃油时间、燃油脉宽及点火提前角等。

其他的控制单元工作原理同发动机电控系统类似，车身上的这些控制单元并不是独立工作，而是作为汽车这样一个整体，它需要信息的共享，因此存在一个信息传递的过程。CAN总线就是把以前一线一用的专线制改成一线多用制，这样可以大大减少汽车上电线的数量，也可以简化整个汽车的布线。在形成OBD-II标准之前，各厂商的汽车诊断座均不相同。比如奔驰为38脚，美洲车12脚，欧宝为10脚，丰田为17脚，宝马为20脚，本田为3脚，澳大利亚西亚款6脚等。

各种各样的诊断插头

各种类型的诊断协议

由于各厂商汽车诊断座与诊断协议差别巨大，导致汽车发生故障时其类型难以判定。因此，SAE（美国机动车工程师学会）制定了OBD-II标准，实行标准检测程序，并且具有严格的排放针对

性，用于实时监测汽车尾气排放情况。

OBD-II标准诊断座

在CAN总线形成标准协议前，欧洲和大部分亚洲汽车采用ISO9141协议，美国通用汽车（GM）使用SAE J1850 VPW通讯协议电路，福特（FORD）汽车则采用SAE J1850 PWM通讯协议电路。可以看出此前以KWP2000协议居多，KWP2000协议最初是基于K线的诊断协议，但由于K线的物理层和数据链路层在网络管理和通讯数据上的局限性，K线逐渐无法满足日益复杂的车载诊断网络的需求。而CAN总线由于其非破坏性的网络仲裁机制、较高的通讯数据和灵活可靠的通讯方式，在车载网络广受青睐，绝大多数车场目前都是采用CAN总线的方式。所以基于K线的KWP2000物理层和数据链路层的协议逐步被基于CAN线的KWP2000协议，也就是ISO15765协议所取代。

K线与CAN总线的比较

K线通信速率较慢，最大达10.4 Kbit/s。此前有一款老款的宝马车是115200bit/s的波特率，但是这种非常罕见。CAN线通信速率可达到1 Mbit/s，其中以500 Kbit/s和250 Kbit/s为主。在传输方面，K线采用单线传输的方式，以字节为单位进行传输，CAN线则以差分信号传输的方式，以CAN帧为单位进行传输。在出错处理中，K线必须要由开发者，如tester，ECU等来完成相应的仲裁处理。而CAN线在物理层面具有完善的通信错误处理机制和总线仲裁机制。报文长度方面，K线最大255个字节，CAN线则是4096字节（指数数据域）。

物理层基于K线

对于OSI分层中的应用层，它的增强型的诊断是基于ISO 14230-3协议，此外还有一些由厂商自行进行定义。而与排放相关的诊断是基于ISO 15031-5协议。

ISO14230协议

KWP2000协议的数据结构由帧头、数据字节和校验和三部分组成。帧头包括起始字节、原码、目的地址和帧长，数据字节和帧校验和位于数据字节，并占2个字节，最后是要加校验和，这三部分

赞同 164 42 条评论 分享 收藏 喜欢 收起

如何进行汽车 CAN 总线开发？

KWP2000协议由于是以字节为单位，所以会涉及到时间。从下图中可以看出它共有四个时间：Tester的字节时间，ECU 1与ECU 2的字节时间，以及ECU收到Tester之后的帧间隔时间。

关注问题 写回答

相关推荐

- 

AUTOSAR 规范与车用控制器软件开发
165 人读过 [阅读](#)
- 

汽车总线故障检修与典型案例
2 人读过 [阅读](#)
- 

汽车车载网络技术详解 第3版
17 人读过 [阅读](#)

一般情况下，P4Tester的字节时间一般默认是20ms左右。P2表示ECU间隔Tester的时间，一般默认是50ms。ECU的P1字节时间默认也是20ms。假设Tester发送8个字节数据，ECU也回复8个字节，P2的间隔时间是50ms，那么获得一帧数据需要320ms的时间，这样获取汽车ECU的数据时间就比较长，效率比较低下。

物理层基于CAN线

基于CAN线的OSI分层结构，其增强型的诊断是基于ISO 14229-1协议与ISO 15765-3协议。而它的排放标准与K线相同，都是基于ISO 15031-5协议。

ISO 15765协议

汽车CAN总线概述

简介

CAN总线最初并不是用于功能开发，而是一种串行的通讯方式，它有着很高的安全等级。特性：

1. 报文可区分优先级，通过ID号进行区分；
2. 保证延迟时间；
3. 设置非常灵活；
4. 时间同步的多点接收；
5. 系统内数据的连贯性；
6. 多主机（任何一个节点都可以当主机）；
7. 错误检测和错误标定；
8. 只要总线一处于空闲，它就将自动破坏的报文重新进行传输。
9. 可以将节点的暂时性错误和永久性错误区分开来。

差分信号采用双角线CAN H和CAN L

汽车CAN总线网络拓扑结构

从下图总线中可以看到，在总线中一般挂有很多的ECU，相关的ECU就近采集相关传感器的数据，通过总线来进行传输，这样就可以大幅减少汽车整车的布线。

报文传输

报文的传输中帧格式分为11位识别符的标准帧与29位识别符的扩展帧。帧类型则分为数据帧、远程帧、错误帧和过载帧，这里主要讨论数据帧。

报文滤波的作用是通过设置硬件的滤波器从而过滤不需要被接受的CAN报文。汽车上的节点非常多，但对于每个节点来讲并不是需要所有的数据。比如在ABS系统并不需要掌握发动机的参数信

息，此时就可以通过设置滤波器而过滤掉这些数据，起到滤波器的作用。

数据帧由七个不同的位场组成：帧起始（Start of Frame）、仲裁场（Arbitration Frame）、控制场（Control Frame）、数据场（Data Frame）、CRC场（CRC Frame）、应答场（ACK Frame）、帧结尾（End of Frame）。仲裁帧可以用来区分该帧是11位的标准帧还是29位的扩展帧，以及是数据帧还是远程帧。

基于UDS的CAN总线ISO 15765协议

Network layer protocol data units

网络层的ISO15765协议最多可以发送或接收4096个字节。

从上图中可以看到PDU有四组数据类型，分别是单帧的SF N_PDU，首帧的FF N_PDU，连续帧的CF N_PDU和流体控制帧的FC N_PDU。

单帧第一个字节的高四位是0，低四位表示帧长，即这一帧有几个字节的有效数据。

首帧第一个字节的高四位是1，第一个字节低四位与第二个字节一同组成帧长，所以最大的是4095个字节。

连续帧第一个字节的高四位是2，SN表示计数器。

流体控制帧第一个字节的高四位是3，低四位是FS（FlowStatus），当FS为0的时候表示继续发送，1表示等待，2则表示溢出。BS为0表示ECU或Tester可以持续回复连续帧数据，BS值非0表示回复连续帧的帧间隔之间可以插入一帧流体控制帧。STmin可以控制连续帧的帧间隔时间，当一些控制器速度达不到要求时，可以通过控制连续帧的帧间隔时间，使其存储到缓冲器中，进行处理。

CAN的UDS

基于CAN的UDS，就是基于ISO 14229-1协议。从下图中可以看到ISO 14229-1协议有不同的Service Id，可以支持不同的服务需求。

比如10表示可进行Diagnoistic Session Control操作，11表示可以对ECU进行Reset操作，27表示可进行Security Access操作等等。

此外还有一些sub-function，不同的sub-function支持不同的功能。比如说Security Access由两帧组成，假设01表示是访问seed，获取种子，在获取到种子之后发送，获取到解析的种子协议再返还回去才能够安全访问。

22表示读取数据信息，23表示可进行Memory By Address操作，19是读取故障码信息，14是清除故障码信息，还可以对ECU进行上传、下传数据等各种操作。

ISO 15031-5协议

基于排放标准的OBD协议，排放标准是ISO 15031协议共有七个章节，每个章节定义不同的内容，主要是以第五章为主。

排放标准的ISO 15031-5协议共有九个服务，不同的服务有不同的功能，在下图均都描述。

我们以Service\$01: Request Current Powertrain Diagnostic data (请求动力系当前数据)来解析ISO 15031-5协议。

每一个PID表示一个传感器，也可以说表示一个数据流，数据流既有状态值，也有数据值。但并不是所有的车都支持所有的PID。首先访问00，如果ECU得到的数据用bit的形式来表示，如果是1表示支持当前的PID，0则表示不支持当前的PID。

所有基于排放标准的ECU，均支持Service\$01，即请求动力系当前数据的D00组数据，也就是说如果想获取相关00组的数据，绝大部分车是支持的。

在获取支持有效PID后，就可以访问实际的PID值，每一次可以访问六个PID。

上图表示00组可以支持01—20组的PID，20可以支持21—40组的PID，以此类推。

上图表示发送0100后ECU的回复。回复41+40，01+40或41表示当前请求动力系当前数据，00表示访问PID获取有效的PID。可以看到第一个字节BF是10111111，即当前ECU支持PID的数据为01及03—08，这7个PID数据，以下类似。

在获取到有效的PID后，我们需要进一步读取当前PID的实际内容，每次最多可以读取六个PID值。15表示Bank 1 – Sensor 2，01表示当前的故障灯状态及故障码数量；05表示冷却液温度等等，如上图所示。

回复值的顺序与初始发送的数据并不一定相同，但是会先发送PID的组号，再补充PID当前组号的实际内容。通过上图可以知道，41表示请求动力帧回复；05表示冷却液温度；61表示冷却液温度的实际值，其他类似。

15031-5协议的Mode 1

上图为Mode 01中PID的一部分。第一列表示PID的序号，后面有不同数据流内容的描述。比如第一个是前文提到的故障码状态与故障码数量，PID2表示冻结的故障码，第三个是Fuel system status。12表示发动机转速，对于转速的最小值、最大值、单位、公式都有具体的规定。

汽车诊断协议的获取通常有三种方式获取骑车诊断协议：

1. ISO 15031-5，由于目前所有车都支持ISO 15031-5协议，所以通过ISO 15031-5协议可以获取大部分的数据信息；
2. 厂家原厂协议，适用于部分不在15031-5内，由厂家自行定义的协议；
3. 第三方工具破解诊断协议。当拿不到原厂协议，需要获取的信息也不在ISO 15031-5里时，只能通过第三方工具进行破解，比如原厂诊断仪和综合诊断仪等。

原厂诊断仪：日产Consult诊断仪、丰田GTS诊断仪、本田HDS诊断仪、大众5053/5054诊断仪等。

综合诊断仪厂家：元征、金奔腾、朗仁、道通AUTEL、BOSCH等。

汽车诊断仪破解过程：

Step 1：

利用第3方工具采集实车诊断数据；

Step 2：

编写汽车模拟程序，即模拟汽车ECU和汽车诊断工序进行通讯；

Step 3：

破解诊断协议，目前基本上均为手动破解；

Step 4：

编写汽车诊断程序；

Step 5：

实车测试。

作者简介

邹凤毕业于中科院人工智能学院控制工程专业，研究领域是不同行业CAN总线的运用和发展，具有十三年的嵌入式和CAN总线工作经验。

知乎机构号：Momenta，打造自动驾驶大脑。

基于深度学习的环境感知、高精度地图、驾驶决策技术，让无人驾驶成为可能。

知乎专栏：Paper Reading，集聚自动驾驶知名大咖的前沿知识分享平台，欢迎申请加入或直接投稿。

编辑于 2018-07-18

赞同 35 6 条评论 分享 收藏 喜欢 收起

赞同 164 42 条评论 分享 收藏 喜欢 收起

