

Can协议学习

参考如下:

视频参考:

文档参考:

[CAN详解--协议详解](#)

[CAN总线的标准帧和扩展帧](#)

[CAN总线通信之详解数据帧\(Data Frame\)](#)

[CAN报文格式](#)

学习进度

- ☐ 一、CAN 发展
- ☐ 二、CAN协议基本概念
- ☐ 三、帧
 - []

一、CAN 发展

CAN (Controller Area Network) 即控制器局域网络。由于其高性能、高可靠性、及独特的设计，CAN越来越受到人们的重视。

1、CAN最初是由德国的BOSCH公司为汽车监测、控制系统而设计的。现代汽车越来越多地采用电子装置控制，如发动机的定时、注油控制，加速、刹车控制(ASC)及复杂的抗锁定刹车系统(ABS)等。由于这些控制需检测及交换大量数据，采用硬接信号线的方式不但烦琐、昂贵，而且难以解决问题，采用CAN总线上述问题便得到很好地解决。

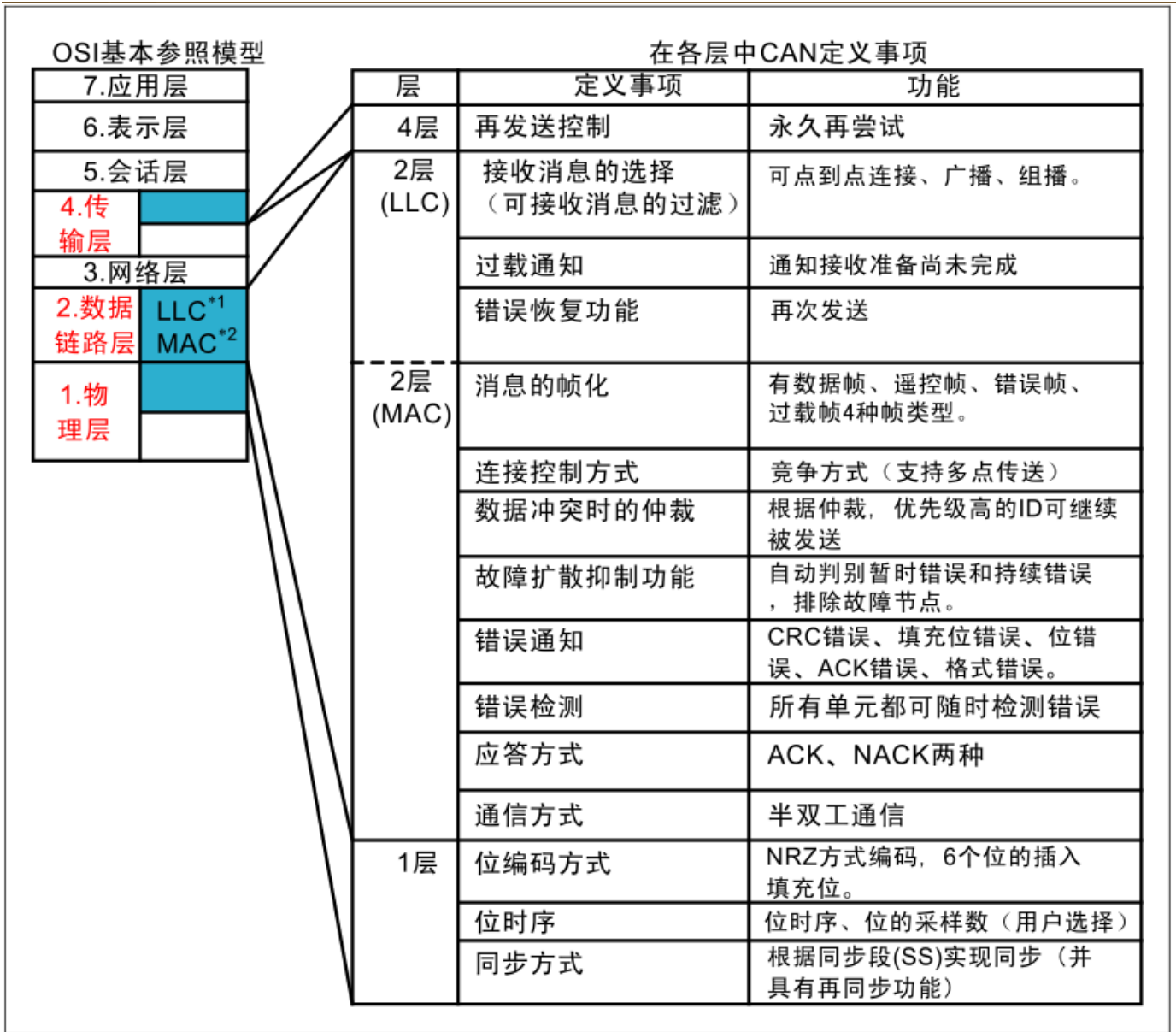
2、1983-1986年 大众与Bosch制定 软件协议，由Intel 生产控制器。 3、1990年 首次应用于汽车 奔驰 S级 12 缸发动机的汽车。 4、1991年9月，NXP半导体公司制定并发布CAN技术规范 CAN2.0A/B，其中CAN2.0A协议规范定义了标准帧格式，CAN2.0B协议规范定义了扩展帧格式。 5、1993年11月，ISO组织正式颁布CAN国际标准ISO11898(高速应用，数据传输速率小于1Mbps)和ISO11519(低速应用，数据传输速率小于125Kbps)。 6、1996年 用于奥迪 A8 D2自动

变速器 3.7升 V8 01V AG5的汽车。 7、1997年 用于帕萨特 B5 AG。 8、1998年 用于宝来、高尔夫 A4 AG。 9、VAN Bus 用于标志、雷诺、雪铁龙等，菲利普公司产品。 10、J1850-HBCC 用于福特，摩托罗拉公司产品。

11、J1850-DLC 用于通用，摩托罗拉公司产品。

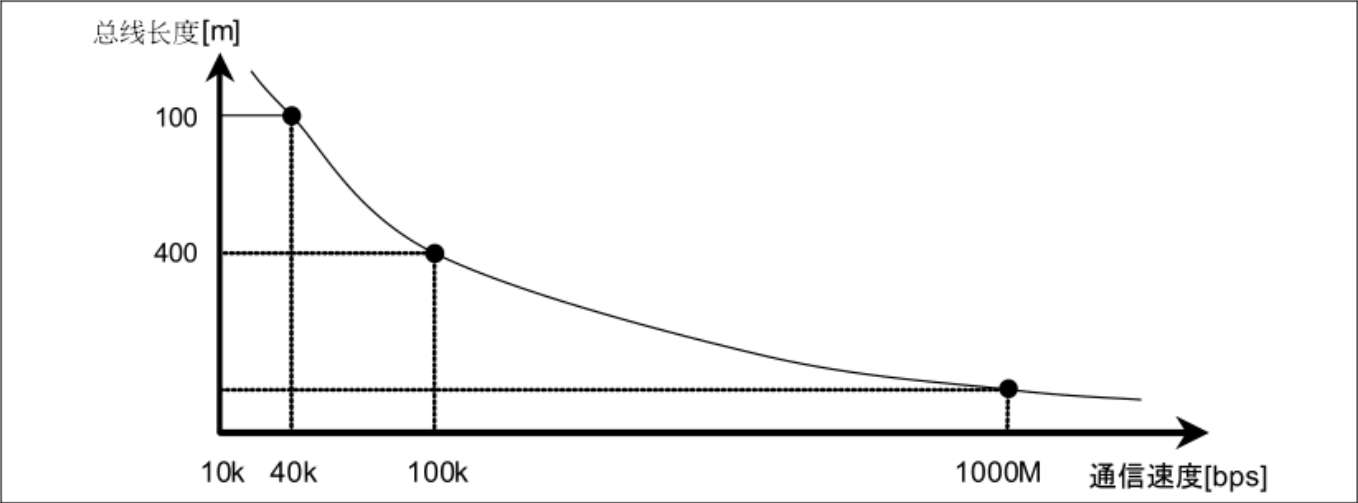
二、CAN协议基本概念

(1) ISO模型与CAN协议



ISO/OSI 基本参照模型和 CAN 协议 https://blog.csdn.net/qq_38880380

(2) 通信速度和最大总线长度



通信速度和最大总线长度

https://blog.csdn.net/qq_38880380

(3) 通信协议分类

电通信 ↑ ↓ 光通信	Class ^{*2}	通信速度	用途	协议
	Class A	~10kbps (车身系统)	灯光类、电动窗、门锁、电动椅、遥控门锁等	● 低速CAN (~125kbps) ● LIN ● 各汽车厂商自有协议
	Class B	10kbps~125kbps (状态信息系统)	电子仪表、驾驶信息、自动空调、故障诊断	● J1850 ● VAN
	Class C	125kbps~1Mbps (实时控制系统)	发动机控制、变速器控制、刹车控制、悬挂控制、ABS等	● 高速CAN (125kbps~1Mbps) ● Safe-by-Wire
	Class D	5Mbps~ (多媒体)		● D2B Optical ● MOST ● IEEE1394

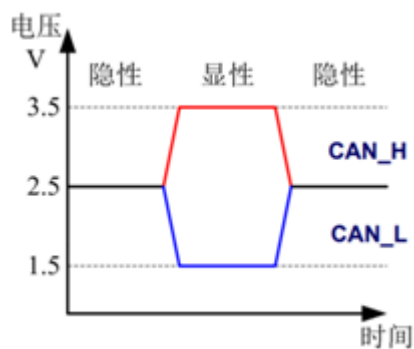
通信协议分类

https://blog.csdn.net/qq_38880380

(4) 显性电平和隐性电平 总线上的电平有显性电平和隐性电平两种。总线上执行逻辑上的“线”与“时”，显性电平的逻辑值为“0”，隐性电平为“1”。“显性”具有“优先”的意味，只要有一个单元输出显性电平，总线上即为显性电平。并且，“隐性”具有“包容”的意味，只有所有的单元都输出隐性电平，总线上才为隐性电平。（显性电平比隐性电平更强。）

(5)CAN帧表达

CAN信号采用差分信号传输，通过差分电压来表达信息。通常情况下只需要两根信号线就可以进行正常的通信。



两条CAN线的差分电压在某个阈值内才是有效的，ISO11898中（高速Can）这个阈值为2V（空闲 2.5V CAN_H 3.5V，CAN_L 1.5V），ISO11519中，这个阈值为3V，电平差变化时间约为2μs

差分电压 = CAN_H – CAN_L

差分电压差范围在2V时表现为显性，定为逻辑0。差分电压差范围在0V时表现为隐性，定为逻辑1。

CAN帧表示为显性标志为有效位。

(6)位填充规则

在CAN发送帧时，为避免总线上产生过多连续相同的电平表达（连续的显性电平或隐性电平），导致使收发双方失步。因此位填充确保最多5个位间就会产生反转电平，确保发送端与接收端进行同步传输。错误帧、过载帧和帧结束之间的连续电平不需要位填充。 位填充适用范围：7帧起始、仲裁段、控制段、数据段、校验段

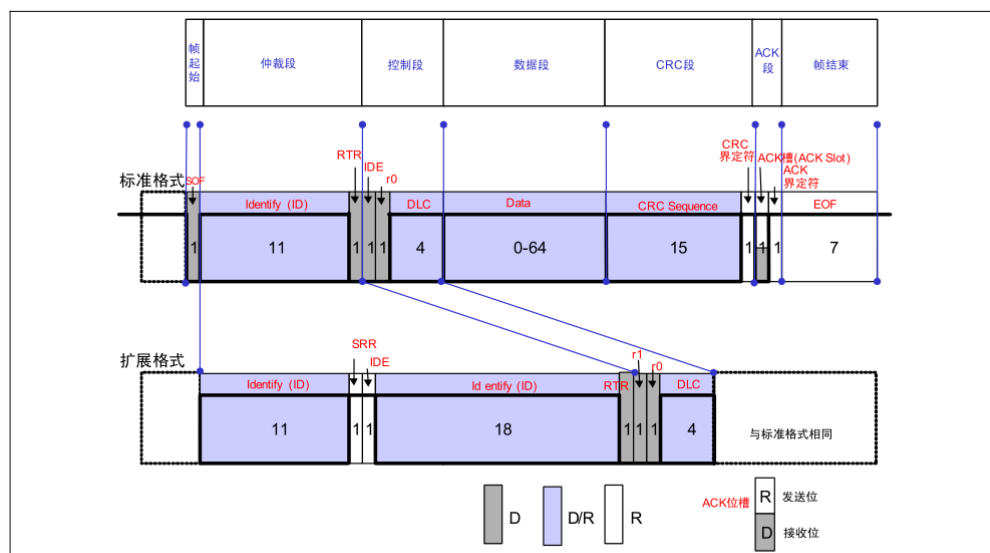
三、帧

CAN协议的报文传输过程中有：

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	用于当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。

https://blog.csdn.net/qq_38880380

(1) 数据帧 A、数据帧的构成



数据帧的构成

https://blog.csdn.net/qq_38880380

数据帧由7个段构成，分别是：(1) 帧起始：1bit，显性信号，表示数据帧开始的段。

扩展帧无帧起始。

(2) 仲裁段：表示该帧优先级的段。

包括两部分：标识符位 (Identifier field-ID) 和远程发送请求位 (Remote Transfer Request-RTR)。

标识符位，是一个功能性的地址 (Functional address)，CAN接收器通过标识符来过滤数据帧。标准格式的数据帧的标识符 (CAN-ID) 长度为11位 (11 bits)，ID10~ID0，ID10为最高权重位 (MSB)，ID0为最低权重位 (LSB)，按照ID10~ID0的顺序进行传输。CAN协议还规定：前7位最高权重位 (ID10~ID4) 不能都为“隐性”信号。

远程发送请求位，简称“RTR”，1 bit。虽然这个位的名字很长，但它的功能很简单，用于区分该帧是数据帧还是远程帧：“显性信号 (0)”代表数据帧 (Data Frame)；“隐性信号 (1)”代表远程帧 (Remote Frame)。

(3) 控制段：表示数据的字节数及保留位的段。

包括6个位 (6 bits)，分别是扩展标识符位 (Identifier Extension bit-IDE, 1 bit)；保留位0 (Reserved bit0-r0, 1 bit)；数据长度编码位 (Data Length Code-DLC, 4 bits)。

扩展标识符位 (IDE)，用来表示该帧是标准格式还是扩展格式；

保留位0 (r0)，保留，以后使用；

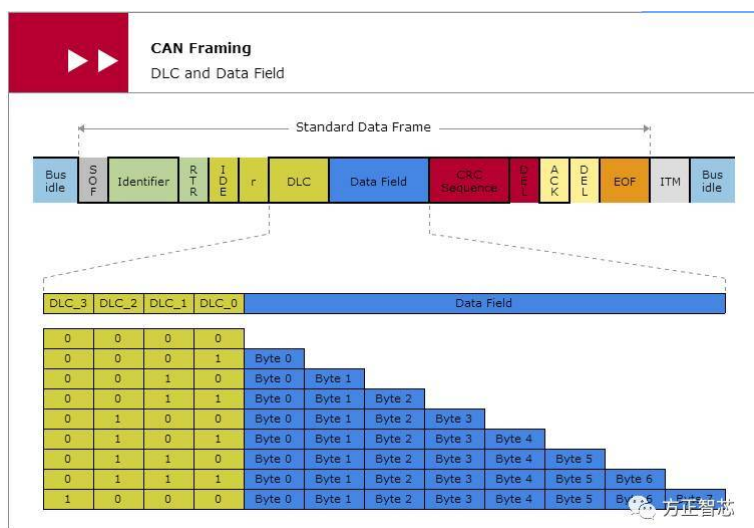
数据长度编码位 (DLC)，4位 (4 bits)，包括DLC3~DLC0，表示该帧实际发送的数据的长度 (以字节为单位)；DLC的编码规则如下：(其中“d”表示显性信号；“r”表示隐性信号；)

Number of Data Bytes	Data Length Code			
	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d	d	d

d 'dominant'
r 'recessive'

(4) 数据段：数据的内容，可发送 0~8 个字节的数据。

发送数据的内容，最多8个字节。数据段与DLC的对应关系如下图：



(5) CRC 段：检查帧的传输错误的段。

包括循环校验序列（CRC Sequence）和界定符（Delimiter, DEL）：

循环校验序列（CRC Sequence），15 bits，用于校验传输是否正确；

界定符（DEL），1 bit，隐性信号，表示循环校验序列的结束；

(6) ACK 段：表示确认正常接收的段。

包括确认位（ACK SLOT）和界定符（Delimiter, DEL），表示确认是否正常接收。

确认位（ACK），1 bit，节点收到正确的CRC序列时，发送端的ACK位被置位；

界定符（DEL），1 bit，隐性信号；

(7) 帧结束：隐性信号，表示数据帧结束的段。

以上是标准格式的数据帧，下面来看看扩展格式与标准格式有哪些不同

(2) 仲裁段：

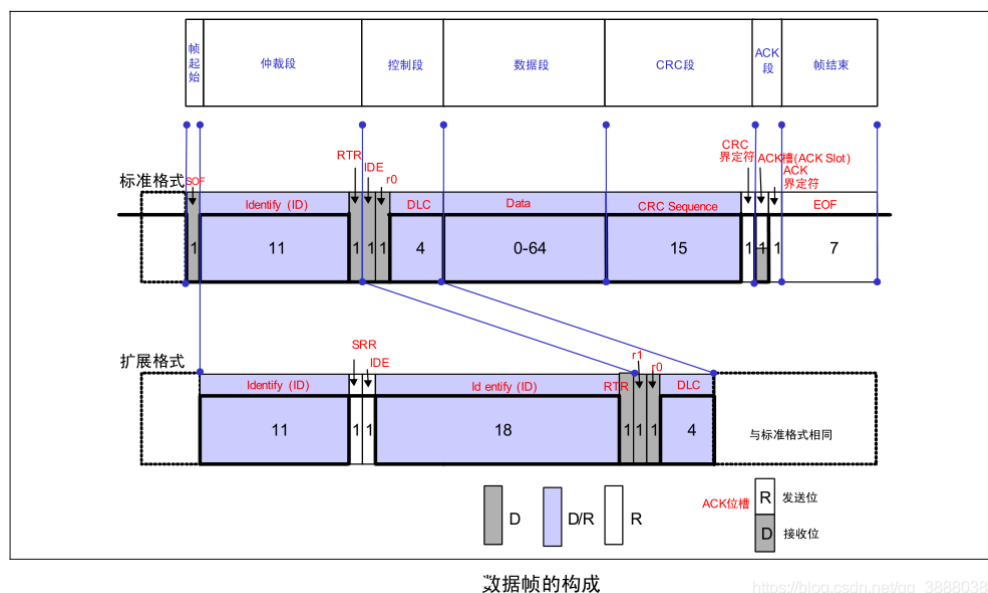
替代远程请求位 (Substitute Remote Request Bit, SRR) , 1 bit, 隐性信号;

扩展位 (Identifier Extension bit, IDE) , 1 bit, 标识扩展帧还是标准帧;

扩展标识ID位 (Extended Identifier, Extended ID) ,18 bits, 存放扩展标识ID;

(3) 控制段:

与标准帧不同的是: 用保留1 (r1) 取代了标准帧的IDE, r0和DLC是一样的; 也就是说, 扩展帧有两个保留位r1 和r0, 这两个保留位均需要发送显性信号;



B、数据帧的标准帧和扩展帧的区别

(1) 标准帧和扩展帧的区别主要决定帧ID的长度

标准帧的帧ID长度是11位, 帧ID的范围是000-7FF。仲裁域是连续的11位, 可以出现 2^{11} 种报文;

扩展帧的帧ID长度是29位, 帧ID的范围是0000 0000-1FFF FFFF。仲裁域有29位, 可以出现 2^{29} 中报文, 且在数据链路上是有间隙的(对操作者透明)。

例如, CANopen帧ID最多是11位的, 因此是标准帧; 而SAE 1939协议通常是18XX XXXX的帧ID, 是29位的, 因此是扩展帧。

控制帧中的DLC(数据长度)完全相同, 但保留位不同, 标准帧IDE、R0, 扩展帧R1、R0, 必须以显性电平发送(由数据链路层操作), 对程序员透明;

其它的都一模一样, 所以, 其实CAN标准帧和扩展帧只是在ID的长度上不一样, 以能扩展更多的CAN节点, 更好地支持上层协议而已。

(2) CAN2.0B标准帧和扩展帧的信息区别

标准帧信息为11个字节(3 + 8)，包括两部分：信息和数据部分。前3个字节为信息部分。

- 字节1为帧信息。第7位（FF）表示帧格式，在标准帧中，FF = 0；第6位（RTR）表示帧的类型，RTR=0表示为数据帧，RTR=1表示为远程帧；DLC表示在数据帧时实际的数据长度。
- 字节2、3为报文识别码，11位有效。
- 字节4~11为数据帧的实际数据，远程帧时无效。

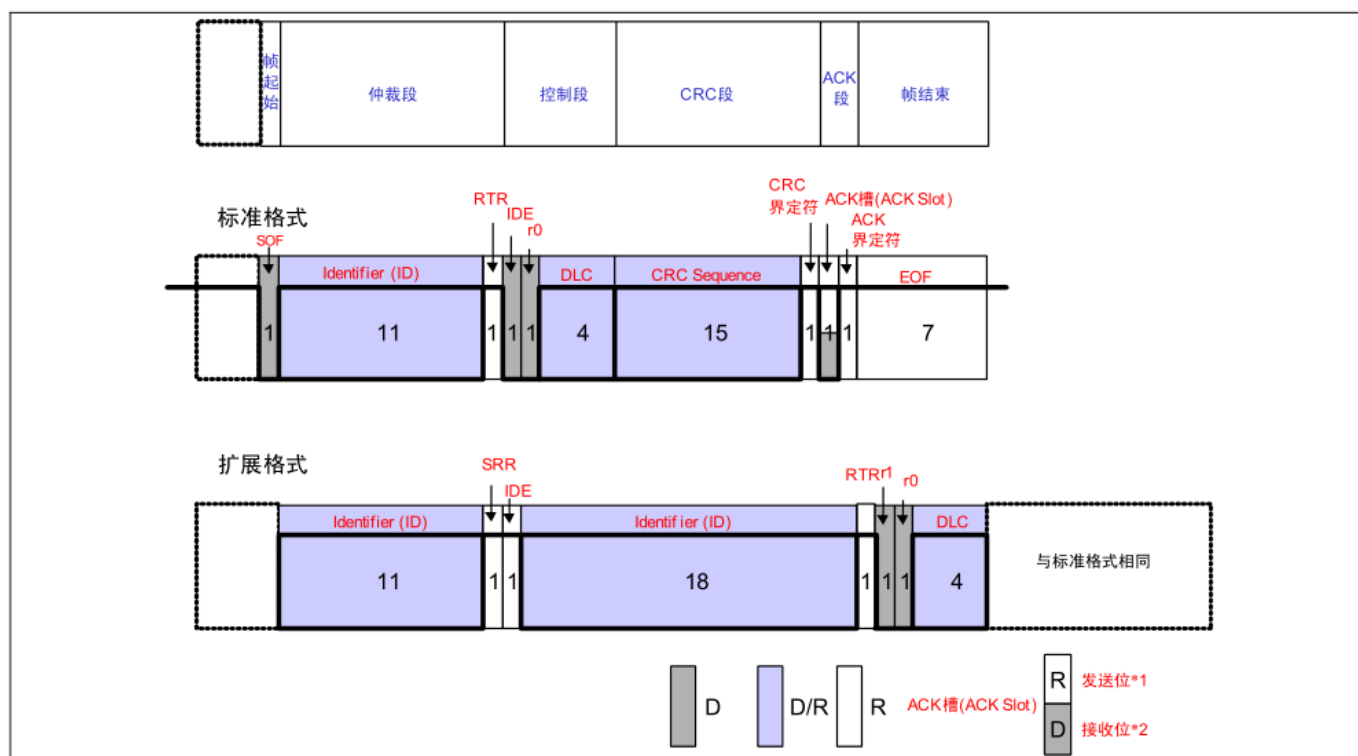
位置	7	6	5	4	3	2	1	0
字节01	FF	RTR	×	×	DLC（数据长度）			
字节02	(报文识别码)				ID.10—ID.3			
字节03	ID.2—ID.0			×	×	×	×	×
字节04	数据1							
字节05	数据2							
字节06	数据3							
字节07	数据4							
字节08	数据5							
字节09	数据6							
字节10	数据7							
字节11	数据8							

- CAN 扩展帧信息为13个字节(5 + 8)，包括两部分，信息和数据部分。前5个字节为信息部分
- 字节1为帧信息。第7位（FF）表示帧格式，在扩展帧中，FF = 1；第6位（RTR）表示帧的类型，RTR=0表示为数据帧，RTR=1表示为远程帧；DLC 表示在数据帧时实际的数据长度。
- 字节2~5为报文识别码，其高29位有效。
- 字节6~13数据帧的实际数据，远程帧时无效

位置	7	6	5	4	3	2	1	0
字节01	FF	RTR	×	×	DLC（数据长度）			
字节02	(报文识别码)				ID.28-ID.21			
字节03	ID.20- ID.13							
字节04	ID.12- ID.5							
字节05	ID.4—ID.0					×	×	×
字节06	数据1							
字节07	数据2							
字节08	数据3							
字节09	数据4							
字节10	数据5							
字节11	数据6							
字节12	数据7							
字节13	数据8							

(2) 遥控帧 A、遥控帧的构成

遥控帧由6个段构成，分别是：(1) 帧起始 (SOF)：表示帧开始的段。(2) 仲裁段：表示该帧优先级的段。可请求具有相同 ID 的数据帧。(3) 控制段：表示数据的字节数及保留位的段。(4) CRC 段：检查帧的传输错误的段。(5) ACK 段：表示确认正常接收的段。(6) 帧结束：表示遥控帧结束的段。



遥控帧的构成

https://blog.csdn.net/qq_38880380

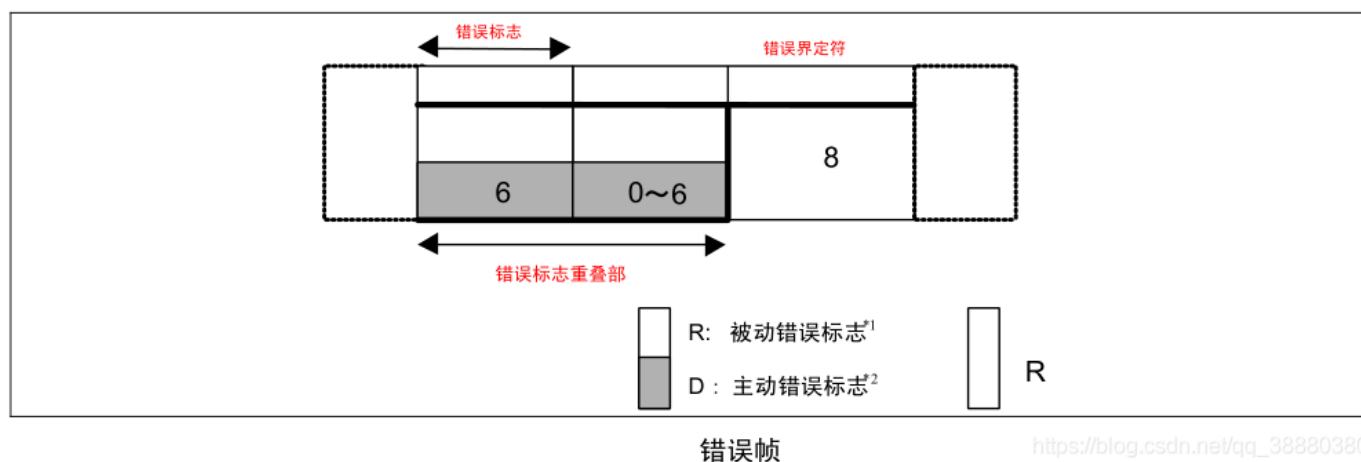
(3) 错误帧 A、错误帧的构成

错误类型：应答错误，填充错误，CRC错误，格式错误

积极错误节点：如果检测到一个错误条件，就会发送积极错误标志。这将引起其它节点检测到填充错误，并开始发送错误标志。因此错误标志叠加在6~12位之间。节点发送完错误标志之后就发送一个隐性位，并监控总线，直到总线上出现一个隐性位，然后再发送7个隐性位。这样一个错误帧就发送完毕了。

消极错误节点：如果检测到一个错误条件，会试图发送一个消极错误标志进行指示。这个消极错误节点会一直等待6个具有相同极性的连续位，等待从消极错误标志起始开始，当检测到6个相同极性的连续位时，消极错误标志发送完成。

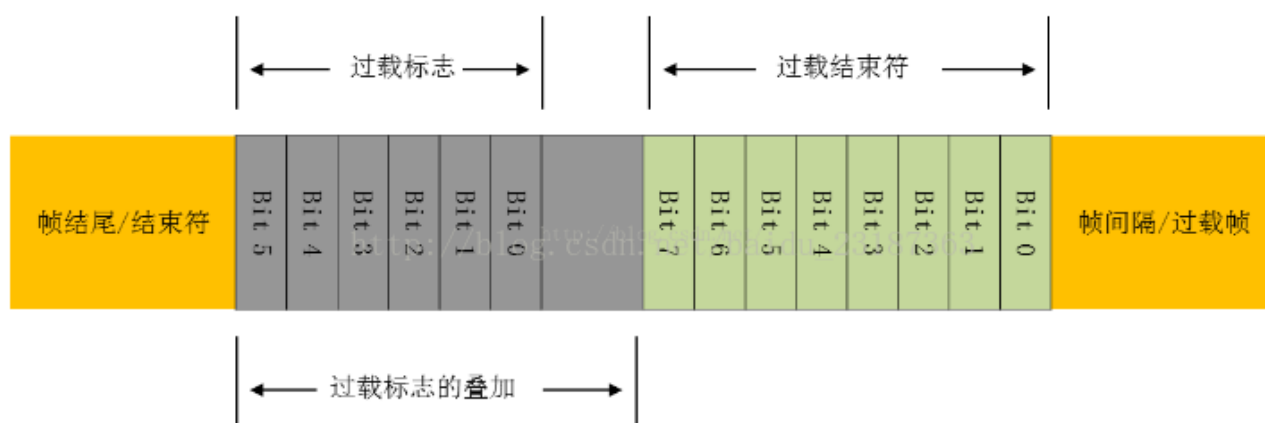
(1) 错误标志 错误标志包括主动错误标志和被动错误标志两种。主动错误标志：6个位的显性位。被动错误标志：6个位的隐性位。(2) 错误界定符 错误界定符由8个位的隐性位构成



(4) 过载帧 A、过载帧的构成

过载标志 6 个位的显性位。过载标志的构成与主动错误标志的构成相同。

过载界定符 8 个位的隐性位。过载界定符的构成与错误界定符的构成相同。



过载条件：接收器要求延迟下一次数据帧或远程帧的到达；在帧间隔间歇场的第一位和第二位检测到显性位；如果CAN节点在错误界定符或过载界定符的第8位采样到一个显性位，则节点会发送一个过载帧，错误计数器不会增加。

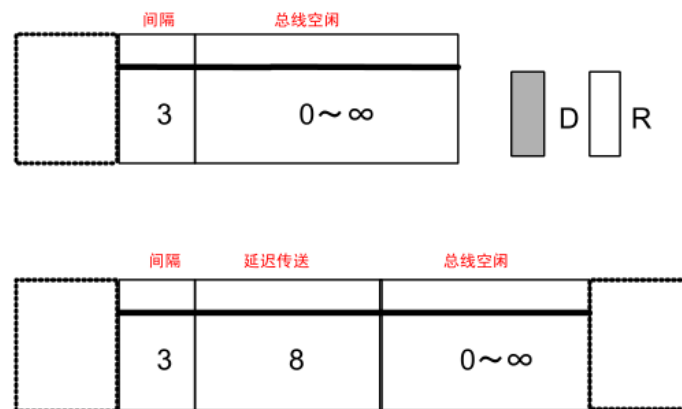
(5) 帧间隔 A、帧间隔的构成

帧间隔是用于分隔数据帧和遥控帧的帧。数据帧和遥控帧可通过插入帧间隔将本帧与前面的任何帧（数据帧、遥控帧、错误帧、过载帧）分开。过载帧和错误帧前不能插入帧间隔。

一) 间隔: 3 个位的隐性位。二) 总线空闲: 隐性电平, 无长度限制 (0 亦可)。本状态下, 可视为总线空闲, 要发送的单元可开始访问总线。三) 延迟传送 (发送暂时停止): 8 个位的隐性位。只在处于被动错误状态的单元刚发送一个消息后的帧间隔中包含的段。

数据帧和远程帧的前面必然有帧间空间。对于主动错误节点和被动错误节点, 帧间空间的结构稍有不同。对于主动错误节点, 帧空间由3个显性位的间歇字段和总线空闲组成。在间歇字段不允许发送数据帧和远程帧。总线空闲的长度任意, 当有显性位时就被认为是帧起始。被

动错误标志除了上边两部分外，在间歇字段后还有8个显性位的挂起传输。在挂起传输阶段被
动错误节点不可以发送数据帧与远程帧。



帧间隔的构成

https://blog.csdn.net/qq_38880380