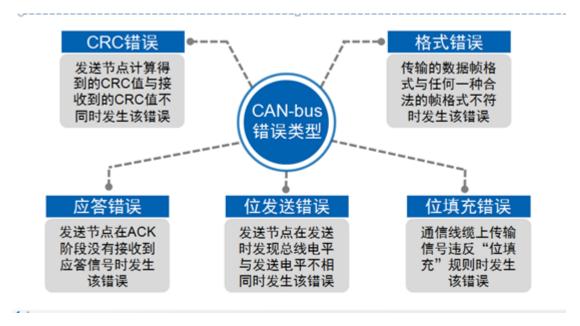
CAN Protocol

CAN基本协议 (https://wenku.baidu.com/view/7c3907ee172ded630a1cb601.html)

CAN特点

- 多主控制,最先访问总线的单元,node可以获得发送权 CSMA/CA,同时发送,看优先级
- 标识符 ID决定优先级 对ID的每个位进行逐个仲裁比较,仲裁获胜的单元继续发送消息,失利的单元立刻停止,转而接收消息。
- 系统柔软性,没有地址概念,随意增加设备
- 可设定通信速率
- 远程数据请求,请求其他单元发送数据。
- 错误检测(边发边收,其他线上设备一直处于接收)错误通知,错误恢复
- 故障封闭
- 连接数量受电气负载限制, 理论上没有约束

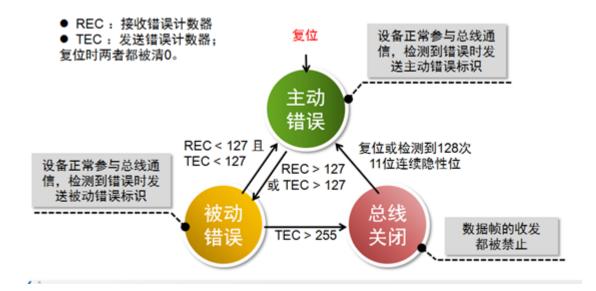
尽管CAN-bus是可靠性很高的总线,但依然可能出现错误; CAN-bus的错误类型共有5种



错误有三类

- 主动错误,可以参加总线通信,输出主动错误标志。
- 被动错误容易引起错误的状态,不能积极的发出错误通知。
- 总线关闭接收,发送被禁止

为防止自身由于某些原因导致无法正常接收的节点一直发送错误帧,干扰其他节点通信, CAN-bus规定了节点的3种状态及其行为如图所示。



这些错误处理的机制是由硬件自主完成的这样做的目的就是只要CAN在收到数据肯定是正确的数据

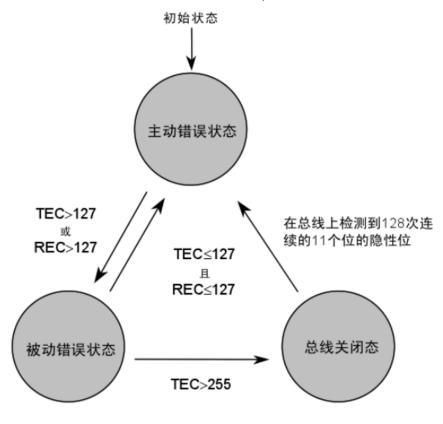
总线关闭态是不能参加总线上通信的状态。

信息的接收和发送均被禁止。

这些状态依靠发送错误计数和接收错误计数来管理,根据计数值决定进入何种状态。错误状态和计数值的关系如表 1 及图 4 所示。

单元错误状态	发送错误计数值(TEC)		接收错误计数值(REC)
主动错误状态	0~127	且	0~127
被动错误状态	128~255	或	128~255
总线关闭态	256~		_

表 1. 错误状态和计数值



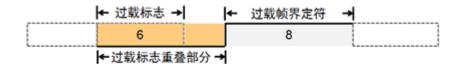
TEC: 发送错误计数值 REC: 接收错误计数值

图 4. 单元的错误状态

过载帧



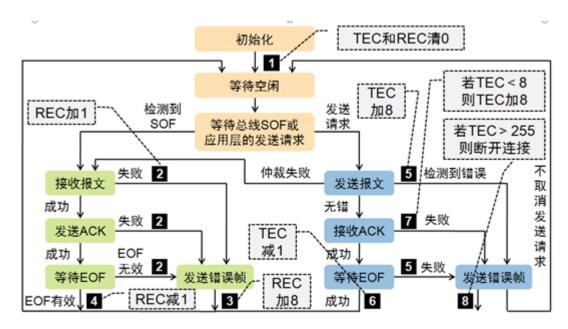
由于存在多个节点同时过载且过载帧发送有时间差问题,可能出现过载标志叠加后超过**6**个位的现象如所示图



帧间隔用于将数据帧或远程帧和他们之前的帧分离开,但过载帧和错误帧前面不会插入帧间 隔。



CAN总线数据链路层处理数据流程图



CAN ISO标准化

ISO11898 ISO11519-2两种



(2) 关于ISO11519

ISO11519 是通信速度为 125kbps 以下的 CAN 低速通信标准。

ISO11519-2 是 ISO11519-1 追加新规约后的版本。

图 6表示 CAN 协议和 ISO11898 及 ISO11519-2 标准的范围。

应用层 表示层 会话层 传输层 网络层	ISO 未对此部分进行标准化。 CAN的再发送控制功能未在 ISO11898/11519-2中标准化。
数据链路层 ISO11898/ISO11519-2 物理层 ISO11898/ISO11519-2不同	数据链路层及物理层的一部分在ISO中进行 了标准化。 对于数据链路层,ISO11898和ISO11592-2 定义的内容相同。 对于物理层,ISO11898和ISO11519-2定义的
CAN 协议定义的部分 CAN 协议中ISO11898/115	内容不同。 19 -2定义标准化的部分

图 6. ISO 标准化的 CAN 协议

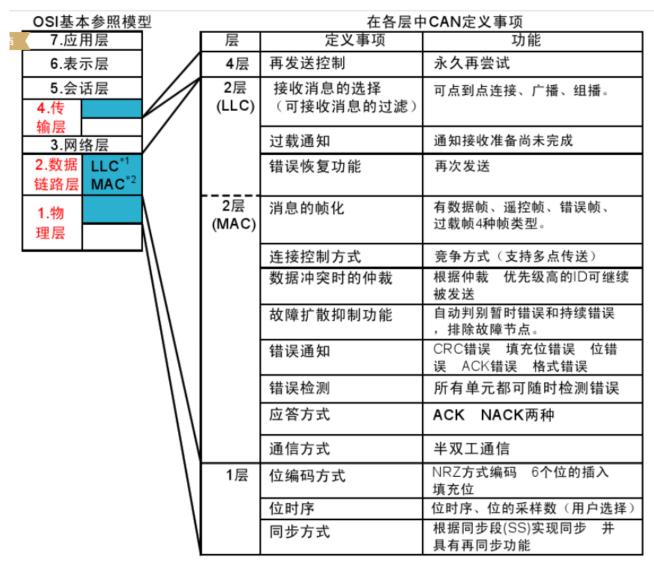
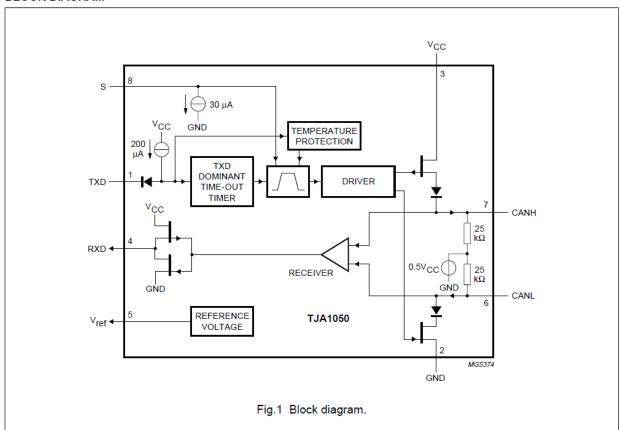


图 5. ISO/OSI 基本参照模型和 CAN 协议

数据链路层分为MAC子层和LLC子层,MAC子层是CAN协议的核心部分,在CAN的硬核控制器里面执行。

A 'TXD dominant time-out' timer circuit prevents the bus lines being driven to a permanent dominant state (blocking all network communication) if pin TXD is forced permanently LOW by a hardware and/or software application failure. The timer is triggered by a negative edge on pin TXD. If the duration of the LOW-level on pin TXD exceeds the internal timer value, the transmitter is disabled, driving the bus into a recessive state. The timer is reset by a positive edge on pin TXD.

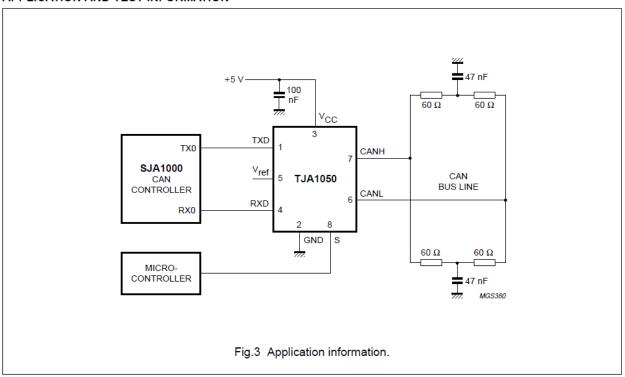
BLOCK DIAGRAM



CANH and CANL voltage absolute value:

Bus lines (pins	S CANH and CANL)					
V _{o(reces)(CANH)}	recessive bus voltage at pin CANH	V _{TXD} = V _{CC} ; no load	2.0	2.5	3.0	V
V _{o(reces)(CANL)}	recessive bus voltage at pin CANL	V _{TXD} = V _{CC} ; no load	2.0	2.5	3.0	V
I _{o(reces)(CANH)}	recessive output current at pin CANH	-27 V < V _{CANH} < +32 V; 0 V < V _{CC} < 5.25 V	-2.0	_	+2.5	mA
I _{o(reces)(CANL)}	recessive output current at pin CANL	-27 V < V _{CANL} < +32 V; 0 V < V _{CC} < 5.25 V	-2.0	_	+2.5	mA
V _{o(dom)(CANH)}	dominant output voltage at pin CANH	V _{TXD} = 0 V	3.0	3.6	4.25	V
V _{o(dom)(CANL)}	dominant output voltage at pin CANL	V _{TXD} = 0 V	0.5	1.4	1.75	V
$ \begin{array}{c} V_{i(dif)(bus)} & \text{ differential bus input voltage} \\ (V_{CANH} - V_{CANL}) \end{array} $		V_{TXD} = 0 V; dominant; 42.5 Ω < R _L < 60 Ω	1.5	2.25	3.0	V
	V _{TXD} = V _{CC} ; recessive; no load	-50	0	+50	mV	

APPLICATION AND TEST INFORMATION



在此之上,不同行业领域应用,有进一步进行了细分。

基于 CAN 的各种标准规格如表 6 所示,如图 10 所示,面向汽车的通信协议以通信速度为准进行了分类。

表 6. CAN 协议和标准规格

名称	波特率	规格	适用领域
SAE J1939-11	250k	双线式、屏蔽双绞线	卡车、大客车
SAE J1939-12	250k	双线式、屏蔽双绞线、12V	农用机械
		供电	
SAE J2284	500k	双线式、双绞线(非屏蔽)	汽车
			(高速: 动力、传动系统)
SAE J24111	33.3k、83.3k	单线式	汽车
			(低速: 车身系统)
NMEA-2000	62.5k、125k、250k、	双线式、屏蔽双绞线	船舶
	500k、1M	供电	
DeviceNet	125k、250k、500k	双线式、屏蔽双绞线	工业设备
		24V 供电	
CANopen	10k、20k、50k、125k、	双线式、双绞线	工业设备
	250k、500k、800k、1M	可选 (屏蔽、供电)	
SDS	125k、250k、500k、1M	双线式、屏蔽双绞线	工业设备
		可选 (供电)	

CAN协议

o. CAN 协议

8.1 帧的种类

通信是通过以下5种类型的帧进行的。

- 数据帧
- 遥控帧
- 错误帧
- 过载帧
- 帧间隔

另外,数据帧和遥控帧有标准格式和扩展格式两种格式。标准格式有 11 个位的标识符(Identifier: 以下称 ID),扩展格式有 29 个位的 ID。

各种帧的用途如表 7 所示,各种帧的构成如图 11~图 15 所示。

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。	

位时序

同步段SS

传播时间段PTS

相位缓冲段1 (PBS1)

相位缓冲段2 (PBS2)



表 11. 段及其作用

段名称	段的作用	Tq 数	
同步段	多个连接在总线上的单元通过此段实现时序	1Tq	8~
(SS: Synchronization Segment)	调整,同步进行接收和发送的工作。由隐性电		25Tq
	平到显性电平的边沿或由显性电平到隐性电		
	平边沿最好出现在此段中。		
传播时间段	用于吸收网络上的物理延迟的段。	1~8Tq	
(PTS: Propagation Time Segment)	所谓的网络的物理延迟指发送单元的输出延		
	迟、总线上信号的传播延迟、接收单元的输入		
	延迟。		
	这个段的时间为以上各延迟时间的和的两倍。		
相位缓冲段 1	当信号边沿不能被包含于 SS 段中时,可在此	1~8Tq	
(PBS1: Phase Buffer Segment 1)	段进行补偿。		
相位缓冲段 2	由于各单元以各自独立的时钟工作,细微的时	2~8Tq	
(PBS2: Phase Buffer Segment 2)	钟误差会累积起来,PBS 段可用于吸收此误		
	差。		
	通过对相位缓冲段加减 SJW 吸收误差。(请		
	参照图 34)。SJW 加大后允许误差加大,但		
	通信速度下降。		
再同步补偿宽度	因时钟频率偏差、传送延迟等,各单元有同步	1∼4Tq	
(SJW: reSynchronization Jump Width)	误差。SJW 为补偿此误差的最大值。		

应用手册

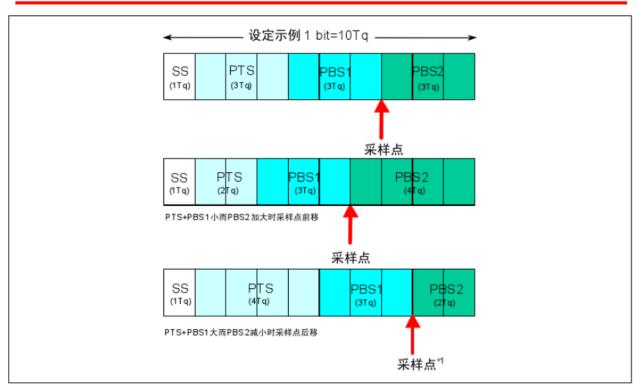
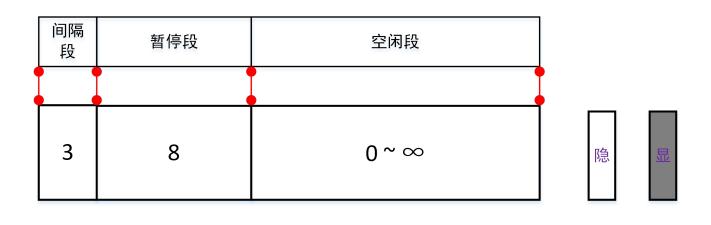


图 32. 1个位的构成

【注】 *1 采样点 所谓采样点是读取总线电平,并将读到的电平作为位值的点。位置在 PBS1 结束处。

间隔段:连续三个隐性位:间隔段期间,所有节点不允许发送数据这或遥控帧,只要在这期 间监听到显性位,接收节点就会发送过载帧。空闲段:连续隐性位,个数不一定,0个或者 多个都可以。总线空闲的时间是任意长的,只要总线空闲,节点就可以竞争总线。 暂停段: 只有处于被动错误状态的节点在发送帧间隔的时候,才会在帧间隔中插入8个连续隐性位的 暂停段。 暂停段,又叫做延迟传送段,为什么节点处于被动状态时会有这样一段呢。原因如 下:首先,考虑主动错误状态的节点Node A,发送主动错误标志之后,随之就要重新发送刚 刚发送失败的报文,但是为了间隔开与前面刚刚发送的错误帧,总线在错误帧之后就会插入 3个隐形位的帧间隔,在这3个隐形位期间,其它的节点不足以判定总线空闲(需要连续11 个隐性位才能判定),所以Node A仍然占据着总线的控制权,于是在帧间隔之后,Node A 能够接着发送报文。现在Node A转入到被动错误状态了,说明它已经不是很可靠了,这个 时候如果没有延迟传送段,在Node A发出被动错误标志之后,它仍然能够在3位的帧间隔之 后立即重新发送报文,这是不符合我们对被动错误状态的处理要求的当然也是不符合CAN协 议的,于是乎对于发送出被动错误标志的节点,总线在帧间隔中加入了8个连续隐性位的延 迟传送段,这样的3+8=11个连续隐性位。就能让Node A在这个帧间隔期间失去对总线的控 制权,从而优先保证其它正常(处于主动错误状态)节点能够使用总线,而不必等着一个已经 不可靠的Node A占据总线。

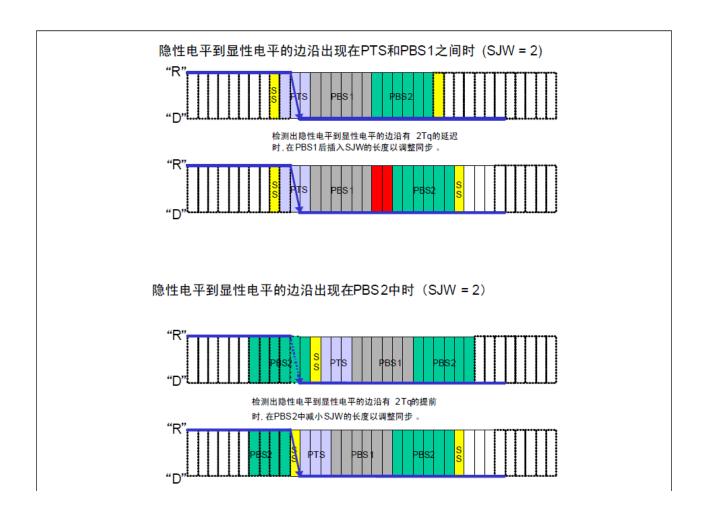


8.14 再同步

在接收过程中检测出总线上的电平变化时进行的同步调整。

每当检测出边沿时,根据 SJW 值通过加长 PB S1 段,或缩短 PBS2 段,以调整同步。但如果发生了超出 SJW 值的误差时,最大调整量不能超过 SJW 值。

再同步如图 34 所示。



再同步,第一种是出现在PTS里面,那么延迟PBS1.

如果出现在PBS2里面,则立刻开始SS

第三种情况,再PBS1里面,则

8.13 硬件同步

接收单元在总线空闲状态检测出帧起始时进行的同步调整。 在检测出边沿的地方不考虑 SJW 的值而认为是 SS 段。 硬件同步的过程如图 33 所示。

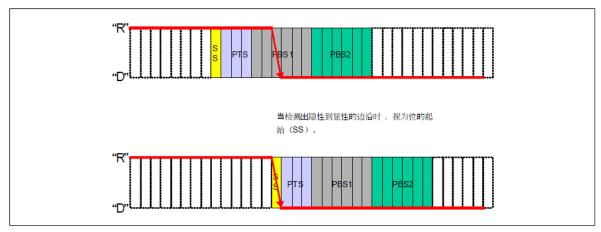


图 33. 硬件同步

8.15 调整同步的规则 硬件同步和再同步遵从如下规则。 (1) 1 个位中只进行一次同步调整。 (2) 只有当上次采样点的总线值和边沿后的总线值不同时,该边沿才能用于调整同步。 (3) 在总线空闲且存在隐性电平到显性电平的边沿时,则一定要进行硬件同步。 (4) 在总线非空闲时检测到的隐性电平到显性电平的边沿如果满足条件(1)和(2),将进行再同步。但还要满足下面条件。 (5) 发送单元观测到自身输出的显性电平有延迟时不进行再同步。 (6) 发送单元在帧起始到仲裁段有多个单元同时发送的情况下,对延迟边沿不进行再同步。

状态机!只有实现CAN控制器才需要实现这些细节,否则都是芯片内部外设已经实现的功能,提供保证。

CANOpen

The basic CANopen device and communication profiles are given in the CiA 301 specification released by <u>CAN in Automation</u>

(https://en.wikipedia.org/wiki/CAN_in_Automation).[1]

(https://en.wikipedia.org/wiki/CANopen#endnote_CiA301) Profiles for more specialized devices are built on top of this basic profile, and are specified in numerous other standards released by CAN in Automation, such as CiA 401[2]

(https://en.wikipedia.org/wiki/CANopen#endnote_CiA401) for I/O-modules and CiA 402[3] (https://en.wikipedia.org/wiki/CANopen#endnote_CiA402) for motion control.

所以目标就是实现兼容CiA401的协议