CAN 控制器根据两根线上的电位差来判断总线电平。发送方通过使总线电平发生变化，将消息发送给接收方。CAN 协议具有以下特点：

1） 。 多主控制。在总线空闲时，所有单元都可以发送消息（多主控制），而两个以上的单元同时开始发送消息时，根据标识符（Identifier 以下称为 ID）决定优先级。ID 并不是表示发送的目的地址，而是表示访问总线的消息的优先级。两个以上的单元同时开始发送消息时，对各消息 ID 的每个位进行逐个仲裁比较。仲裁获胜（被判定为优先级最高）的单元可继续发送消息，仲裁失利的单元则立刻停止发送而进行接收工作。

2） 系统的柔软性。与总线相连的单元没有类似于“地址”的信息。因此在总线上增加单

元时，连接在总线上的其它单元的软硬件及应用层都不需要改变。

3） 通信速度较快，通信距离远。最高 1Mbps（距离小于 40M），最远可达 10KM（速率低于 5Kbps）。

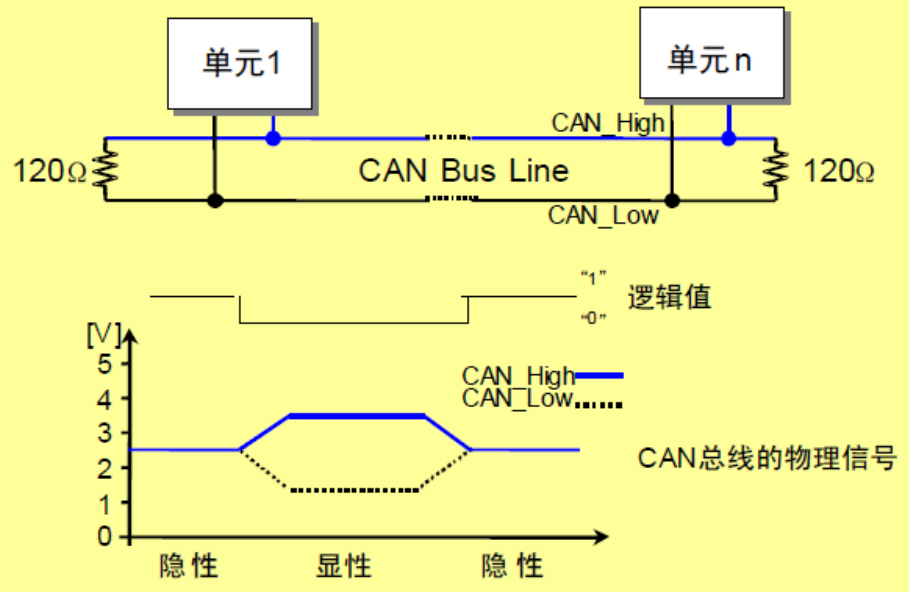
4）具有错误检测、错误通知和错误恢复功能。所有单元都可以检测错误（错误检测功能），检测出错误的单元会立即同时通知其他所有单元（错误通知功能），正在发送消息的单元一旦检测出错误，会强制结束当前的发送。强制结束发送的单元会不断反复地重新发送此消息直到成功发送为止（错误恢复功能）。

5） 。 故障封闭功能。CAN 可以判断出错误的类型是总线上暂时的数据错误（如外部噪声等）还是持续的数据错误（如单元内部故障、驱动器故障、断线等）。由此功能，当总线上发生持续数据错误时，可将引起此故障的单元从总线上隔离出去。

6） 连接节点多。CAN 总线是可同时连接多个单元的总线。可连接的单元总数理论上是没有限制的。但实际上可连接的单元数受总线上的时间延迟及电气负载的限制。降低通信速度，可连接的单元数增加；提高通信速度，则可连接的单元数减少。正是因为 CAN 协议的这些特点，使得 CAN 特别适合工业过程监控设备的互连，因此，越来越受到工业界的重视，并已公认为最有前途的现场总线之一。

CAN协议经过ISO标准化后有两个标准：ISO11898标准和ISO11519-2标准。其中ISO11898

是针对通信速率为125Kbps~1Mbps 的高速通信标准，而 ISO11519-2是针对通信速率为 125Kbps以下的低速通信标准。

ISO11898 物理层特性：

从该特性可以看出，CAN\_H 和 CAN\_L 之差为 2.5V 左右时：为显性电平，对应逻辑 0，。而隐性电平是CAN\_H 和 CAN\_L 之差为 0V，对应逻辑 1。在总线上显性电平具有优先权，只要有一个单元输出显性电平，总线上即为显性电平。而隐形电平则具有包容的意味，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）。另外，在 CAN 总线的起止端都有一个 120Ω的终端电阻，来做阻抗匹配，以减少回波反射。

CAN 协议各种帧及其用途：

|  |  |
| --- | --- |
| 帧类型 | 帧用途 |
| 数据帧 | 用于发送单元向接收单元传送数据的帧（有标准格式和扩展格式） |
| 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（有标准格式和扩展格式） |
| 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 |
| 间隔帧 | 用于将数据帧及遥控帧与前面的帧分离开来的帧 |

由于篇幅所限，我们这里仅对数据帧进行详细介绍，数据帧一般由 7 个段构成，即：

（1） 帧起始。表示数据帧开始的段。

（2） 仲裁段。表示该帧优先级的段。

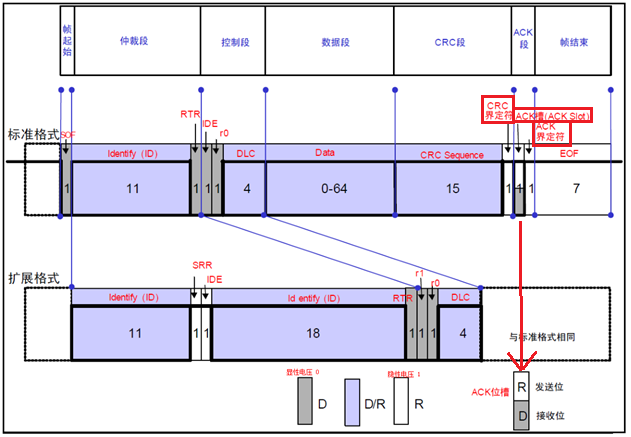
（3） 控制段。表示数据的字节数及保留位的段。

（4） 数据段。数据的内容，一帧可发送 0~8 个字节的数据，MSB。

（5） CRC 段。检查帧的传输错误的段，计算范围包括帧起始、仲裁段、控制段、数据段。

（6） ACK 段。表示确认正常接收的段。

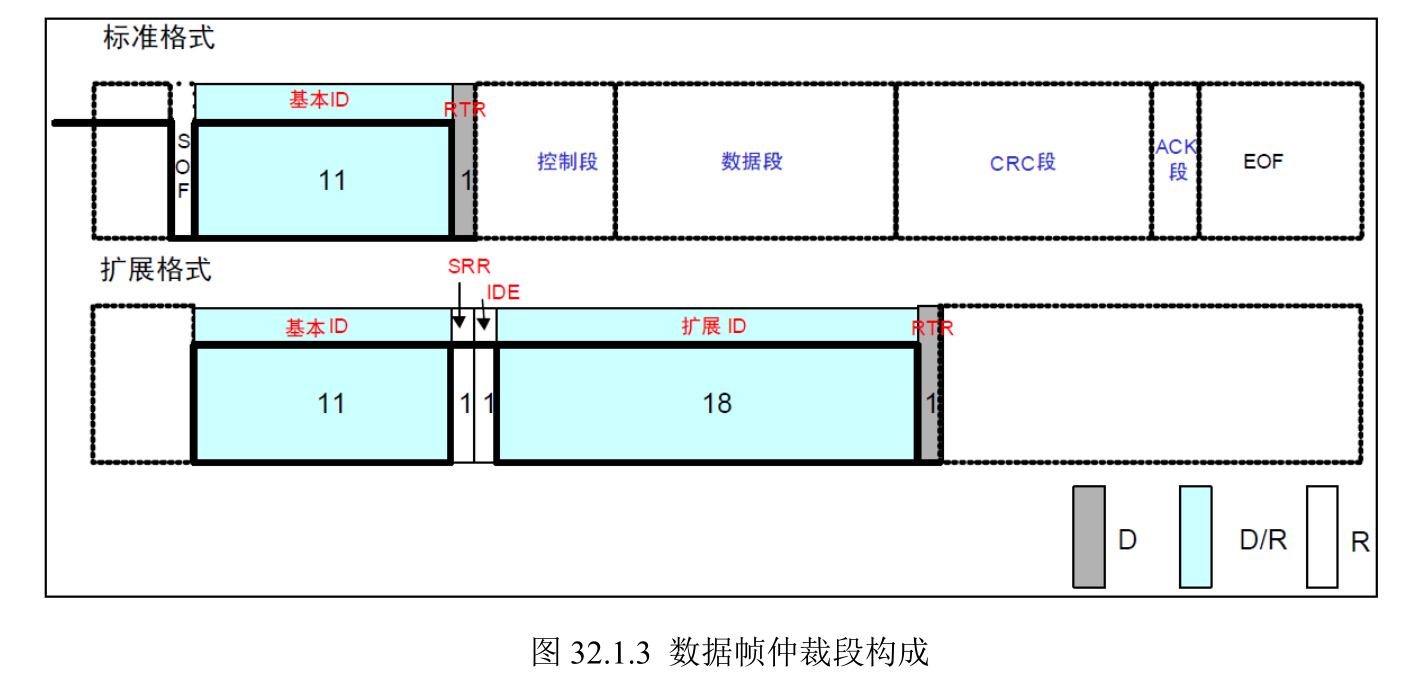
（7） 帧结束。表示数据帧结束的段。



简单说：其中 RTR 位用于标识是否是远程帧（0，数据帧；1，远程帧），IDE 位为标识符选择位（0，使用标准标识符；1，使用扩展标识符），SRR 位为代替远程请求位，为隐性位，它代替了标准帧中的 RTR 位。

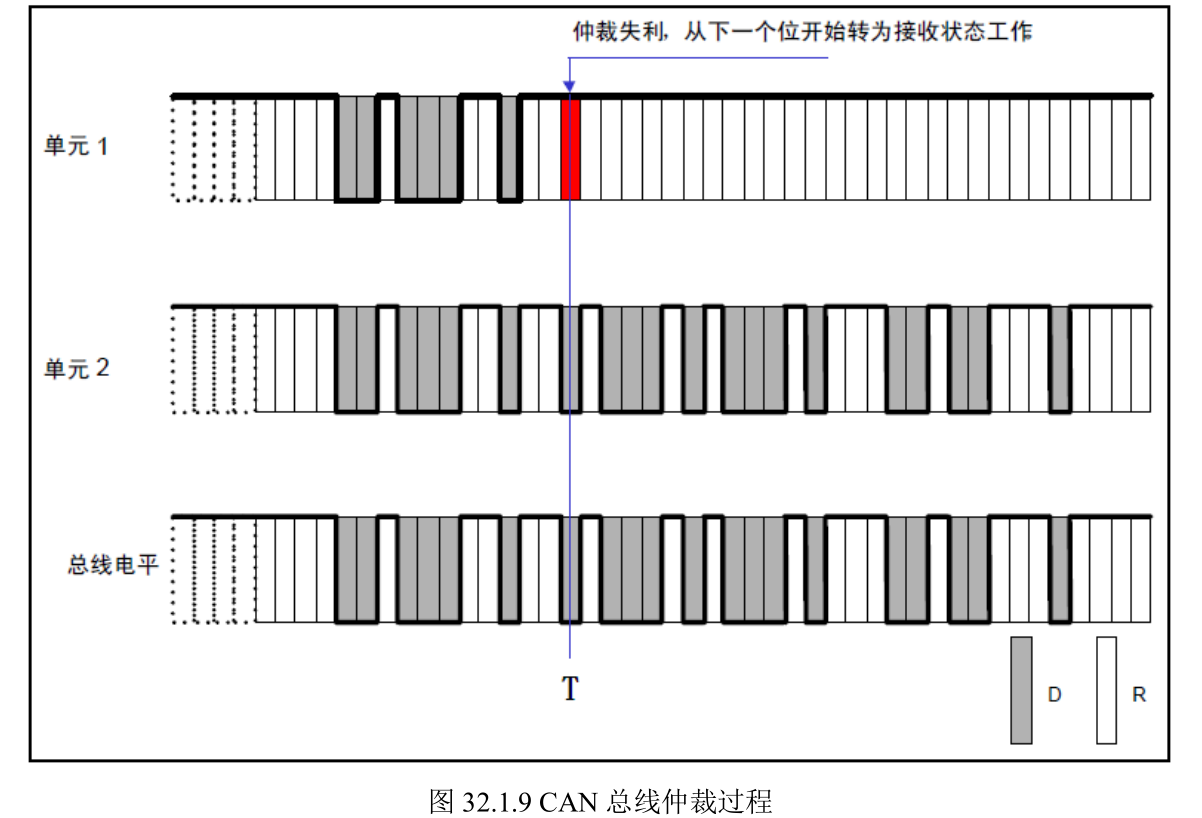
下面分别介绍各个数据段：

仲裁段：



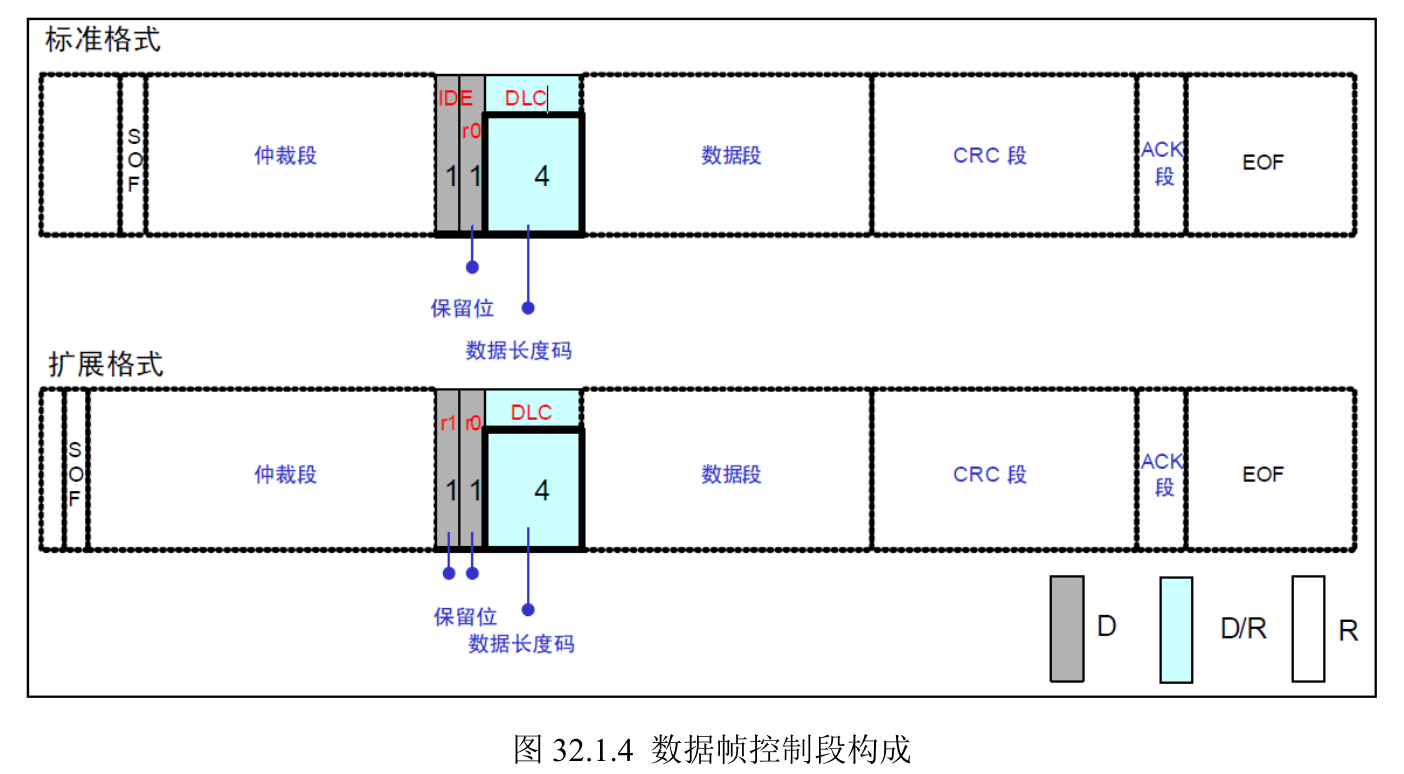
标准格式的 ID 有 11 个位。从 ID28 到 ID18 被依次发送。禁止高 7 位都为隐性（禁止设定：ID=1111111XXXX）。扩展格式的 ID 有 29 个位。基本 ID 从 ID28 到 ID18，扩展 ID 由ID17 到 ID0 表示。基本 ID 和标准格式的 ID 相同。禁止高 7 位都为隐性（禁止设定：基本ID=1111111XXXX）。

其中 RTR 位用于标识是否是远程帧（0，数据帧；1，远程帧），IDE 位为标识符选择位（0，使用标准标识符；1，使用扩展标识符），SRR 位为代替远程请求位，为隐性位，它代替了标准帧中的 RTR 位。



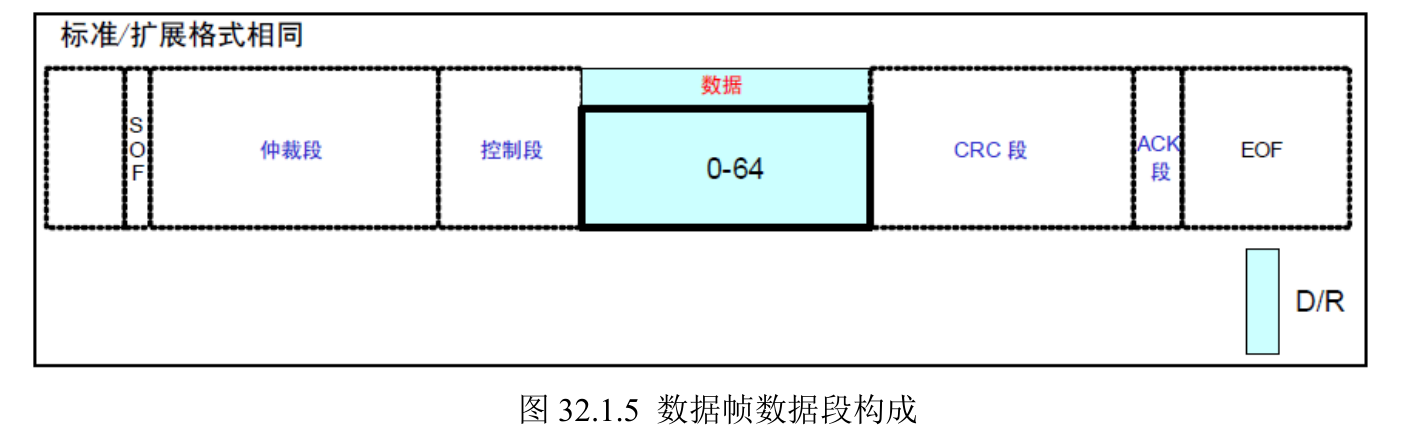
上图中，单元 1 和单元 2 同时开始向总线发送数据，开始部分他们的数据格式是一样的，故无法区分优先级，直到 T 时刻，单元 1 输出隐性电平，而单元 2 输出显性电平，此时单元 1仲裁失利，立刻转入接收状态工作，不再与单元 2 竞争，而单元 2 则顺利获得总线使用权，继续发送自己的数据。这就实现了仲裁，让连续发送显性电平多的单元获得总线使用权。

控制段，由 6 个位构成，表示数据段的字节数：



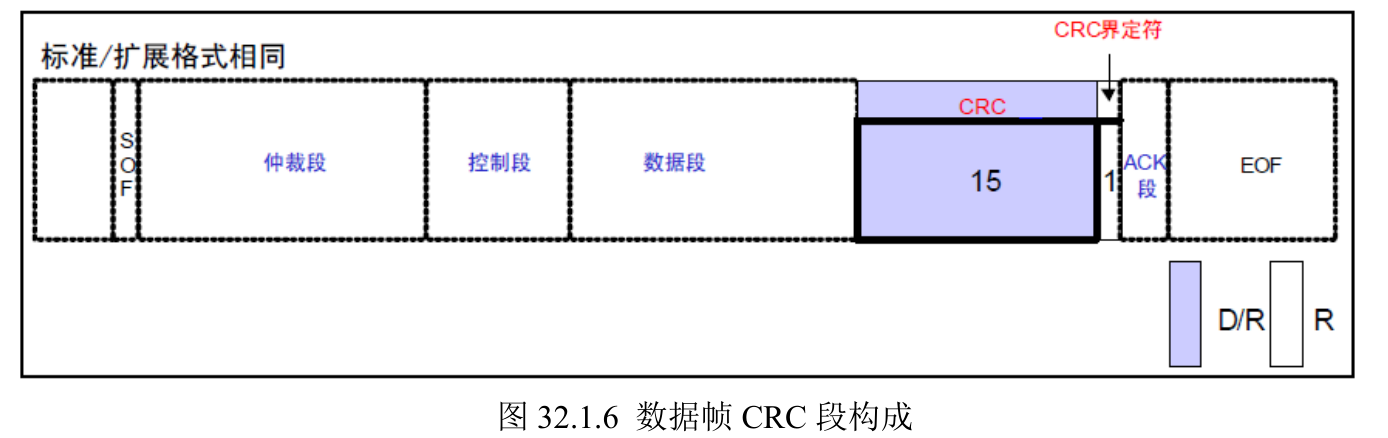
上图中，r0 和 r1 为保留位，必须全部以显性电平发送，但是接收端可以接收显性、隐性及任意组合的电平。DLC 段为数据长度表示段，高位在前，DLC 段有效值为 0~8，但是接收方接收到 9~15 的时候并不认为是错误。

数据段，该段可包含 0~8 个字节的数据，从最高位（MSB）开始输出：



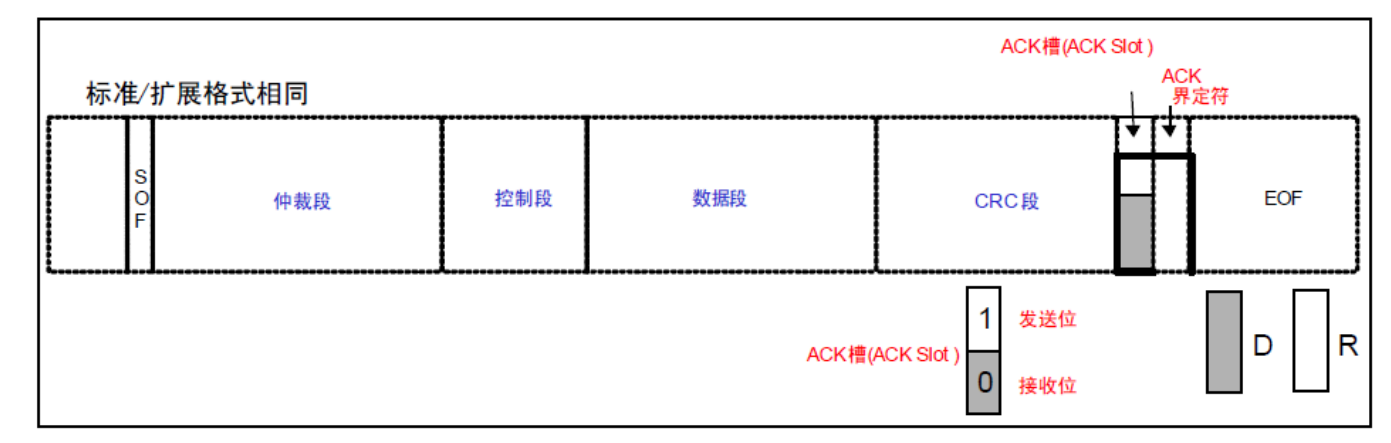
这个段的定义都是一样的

CRC 段，该段用于检查帧传输错误。由 15 个位的 CRC 顺序和 1 个位的 CRC 界定符（用于分隔的位）组成，标准帧和扩展帧在这个段的格式也是相同的：



此段 CRC 的值计算范围包括：帧起始、仲裁段、控制段、数据段。接收方以同样的算法计算 CRC 值并进行比较，不一致时会通报错误。

ACK 段，此段用来确认是否正常接收。由 ACK 槽(ACK Slot)和 ACK 界定符 2 个位组成。标准帧和扩展帧在这个段的格式也是相同的。

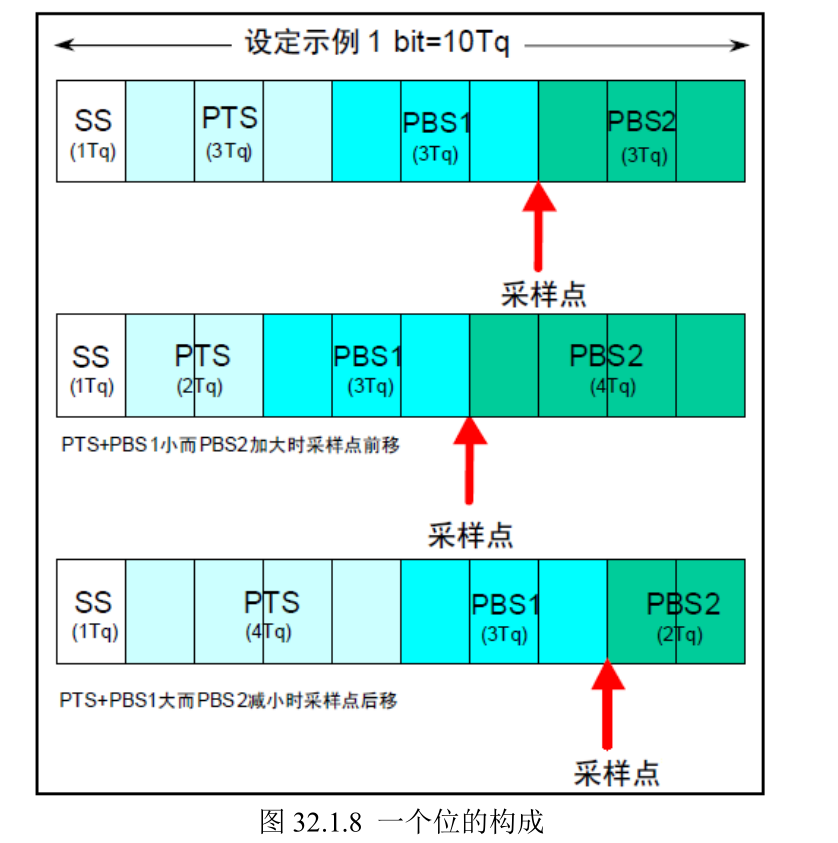


发送单元的 ACK，发送 2 个位的隐性位，而接收到正确消息的单元在 ACK 槽（ACK Slot）发送显性位，通知发送单元正常接收结束，这个过程叫发送 ACK/返回 ACK。发送 ACK 的是在既不处于总线关闭态也不处于休眠态的所有接收单元中，接收到正常消息的单元（发送单元不发送 ACK）。所谓正常消息是指不含填充错误、格式错误、CRC 错误的消息。

帧结束，这个段也比较简单，标准帧和扩展帧在这个段格式一样，由 7 个位的隐性位组成。

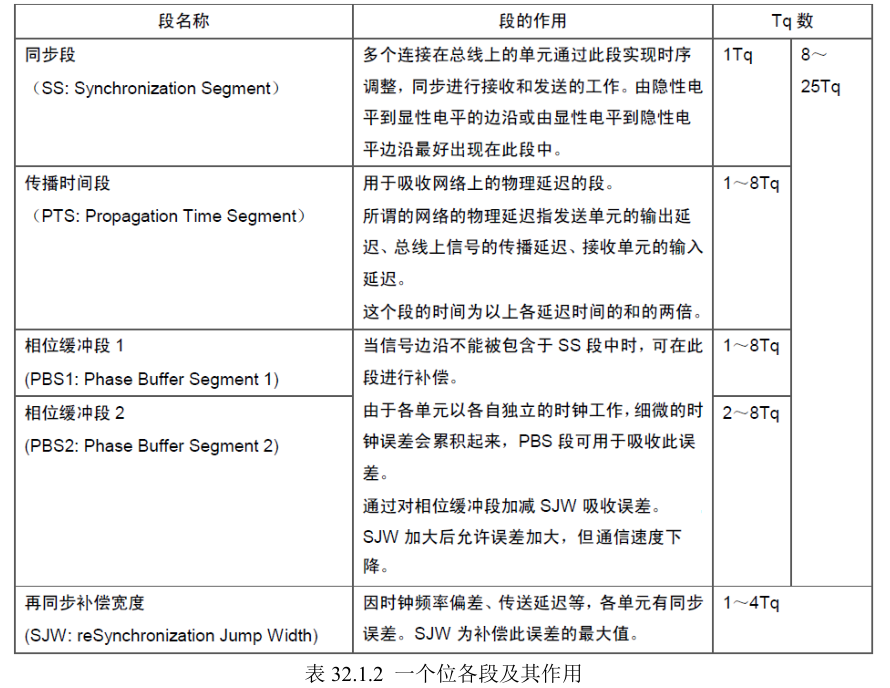
一个数据位的概念：

由发送单元在非同步的情况下发送的每秒钟的**位**数称为位速率。一个**位**可分为 4 段：

同步段（SS）-传播时间段（PTS）-相位缓冲段 1（PBS1）-相位缓冲段 2（PBS2）

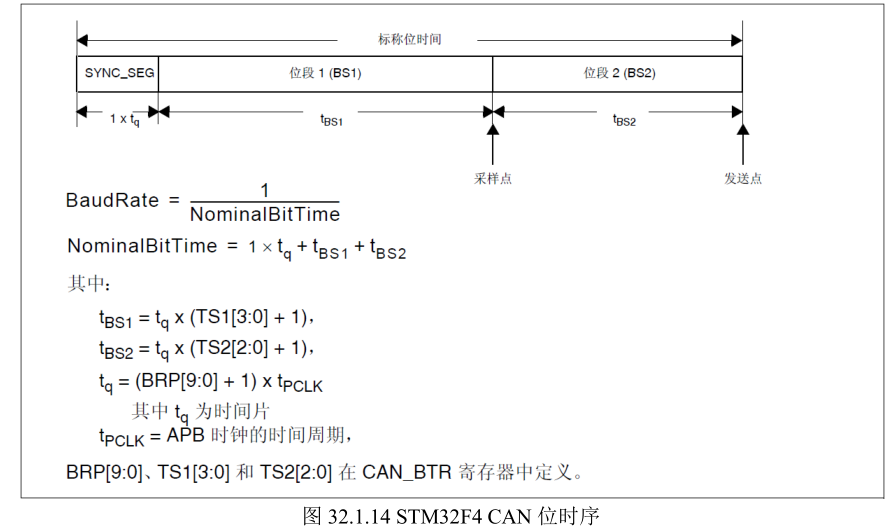
每一个段又由称为 Time Quantum（以下称为 Tq）的最小时间单位构成。

1. 位由多少个 Tq 构成、每个段又由多少个 Tq 构成等，可以任意设定位时序。



STM32F4 把传播时间段和相位缓冲段 1(STM32F4 称之为时间段1)合并了，所以 STM32F4 的 CAN 一个位只有 3 段：

同步段（SYNC\_SEG）-时间段 1（BS1）-时间段 2（BS2）



比如设置 TS1=6、TS2=5 和 BRP=5，在 APB1 频率为 42Mhz 的条件下，即可得到 CAN 通信的波特率=42000/[(7+6+1)\*6]=500Kbps。

(7+6+1)\*6三个数字分别为Tbs1个Tq\ Tbs2个Tq \Tq是几个Tpclk（CAN外设时钟）

在配置库函数时，三段的意义如上图，最好设置成把采样点放在中间的位置，且为500Kbps波特率。

在CAN外设频率为36Mhz时，推荐分频系数(brp+1)为4，三段分别为：1tq\9tq\8tq。

在CAN外设频率为42Mhz时，推荐分频系数(brp+1)为6，三段分别为：1tq\7tq\6tq。

STM32F1 的 bxCAN 的主要特点有：

⚫ 支持 CAN 协议 2.0A 和 2.0B 主动模式

⚫ 波特率最高达 1Mbps

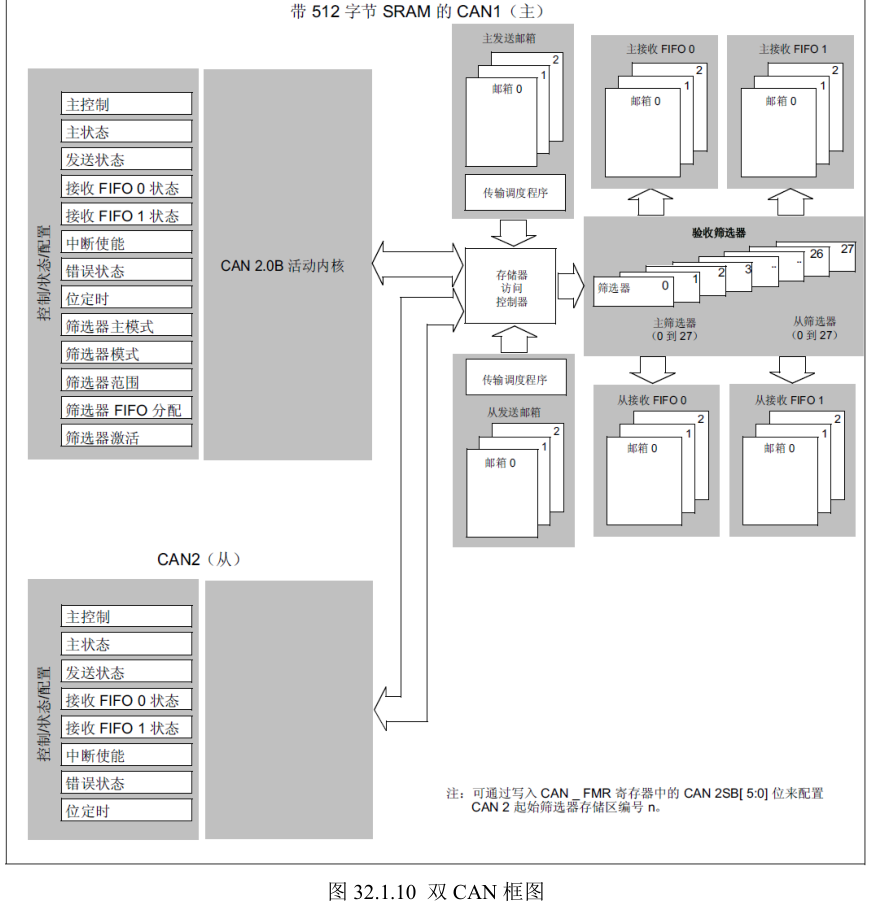
⚫ 支持时间触发通信

⚫ 具有 3 个发送邮箱

⚫ 具有 3 级深度的 2 个接收 FIFO

⚫ 可变的接收过滤器组（最多 28 个），可任意分配给接收FIFO

在 STM32F407ZGT6 中，带有 2 个 CAN 控制器



从图中可以看出两个 CAN 都分别拥有自己的发送邮箱和接收 FIFO，但是他们共用 28 个接收滤波器（用于筛选标识符，即ID，用于只关心收到来自特定ID的信息）。

接收到的信息先进入接收滤波器筛选特定想要的ID的信息，然后存入与其绑定的三级深度的FIFO（0或者1），然后从FIFO中取数据即可

接收滤波器模式：屏蔽位模式（IdMask）和标识符列表模式（IdList）：

举例说明标识符屏蔽位模式（IdMask）：

以下都是概念性阐述，不是真正的写法{设置 CAN\_F0R1=0XFFFF0000，CAN\_F0R2=0XFF00FF00。CAN\_F0R2的0XFF00FF00 就是设置我们需要必须关心的 ID，表示收到的 ID，其为1的位（[31:24]和位[15:8]这 16 个位）必须和 CAN\_F0R1 中对应的位一模一样，而另外的 16 个位则不关心，可以一样，也可以不一样，都认为是正确的 ID，即收到的 ID 必须是 0XFFxx00xx，才算是正确的（x 表示不关心）。

在标准库中：

CAN\_F0R1是CAN\_FilterIdHigh和CAN\_FilterIdLow来配置；

CAN\_F0R2是CAN\_FilterMaskIdHigh和CAN\_FilterMaskIdLow来配置。

在标识符列表模式下：CAN\_F0R1和CAN\_F0R2两个都是指定要接收的标识符ID，接收报文标识符的每一位都必须跟过滤器标识符相同。}

通过配置滤波器来设置硬件接收特定的ID和帧类型的信息，标准库的具体的真正的写法看：STM32CAN过滤器总结.pdf，这个文档里面有能用的例程，只接受数据帧的标准标识符和扩展标识符的代码已经写下，在我的stm32\_framework框架的CAN部分内。