[SDIO 2](#_Toc513199057)

[命令报文 2](#_Toc513199058)

[响应报文 3](#_Toc513199059)

[常规数据包 4](#_Toc513199060)

[宽位数据包 5](#_Toc513199061)

[状态数据块 6](#_Toc513199062)

[SDIO命令 8](#_Toc513199063)

[概述 8](#_Toc513199064)

[基本命令(Class 0) 10](#_Toc513199065)

[面向块的读操作(Class 2) 11](#_Toc513199066)

[面向块的写操作(Class 4) 12](#_Toc513199067)

[擦除命令(Class 5) 13](#_Toc513199068)

[写保护命令(Class 6) 14](#_Toc513199069)

[加锁命令(Class 7) 15](#_Toc513199070)

[特定应用命令(Class 8) 16](#_Toc513199071)

[I/O模式命令(Class 9) 17](#_Toc513199072)

[切换命令(Class 10) 18](#_Toc513199073)

[SDIO响应 19](#_Toc513199074)

[CAN 20](#_Toc513199075)

[CANOpen 23](#_Toc513199076)

[EMCY16位应急错误代码表 25](#_Toc513199077)

# SDIO

## 命令报文

从上到下的顺序是 数据位的由高到低，也是传输的先后的顺序：

|  |  |  |
| --- | --- | --- |
| bit | 名称 | 说明 |
| 47 | 起始位 | 起始位=0  命令的主体包含在起始位与终止位之间； |
| 46 | 传输标志 | 传输标志=1；  表示本报文是命令报文，传输方向是主机🡪SD卡 |
| 45:40 | 命令号 | 000000~111111：对应64种命令 |
| 39:8 | Argument | 每个SD命令报文有32位用于存放该命令的Argument  例如：寻址命令使用这32位来指定目标SD卡的地址、  广播命令使用这32位来指定参数；  如果不使用附加命令，则用0填充 |
| 7:1 | CRC7校验 | CRC校验错误时，SD卡(从机)不执行该命令。 |
| 0 | 终止位 | 终止位=1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 命令报文类型 | 无响应广播命令 | bc | 发送到所有卡，不需要响应 |
| 带响应广播命令 | bcr | 发送到所有卡，同时接收来自所有卡响应 |
| 寻址命令 | ac | 发送到指定卡，DAT线无数据传输  寻址通过使用会话地址来实现，会话地址会在初始化阶段分配给卡。 |
| 寻址数据传输命令 | adtc | 发送到选定卡，DAT线有数据传输 |

## 响应报文

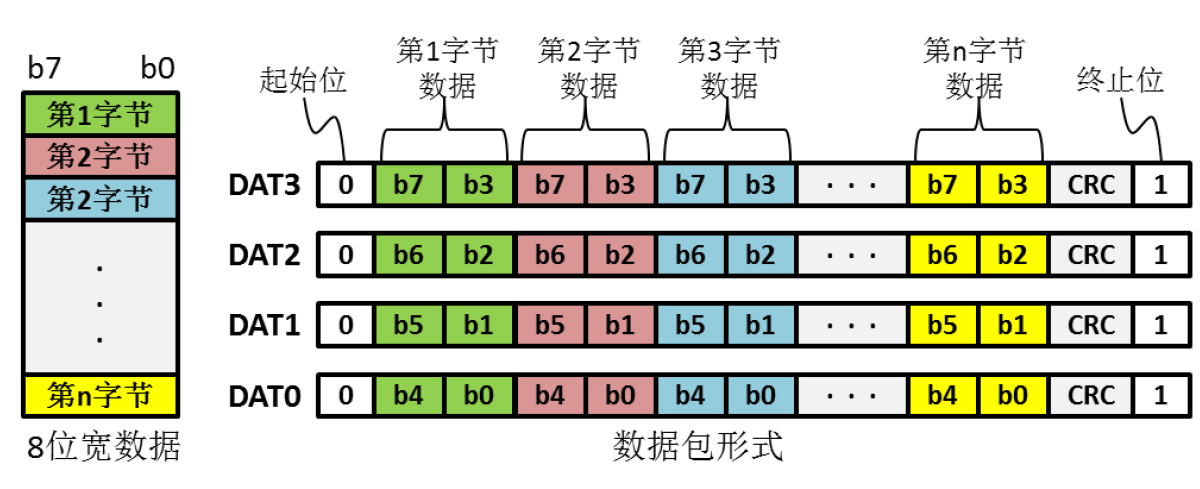
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| R1响应  正常响应 | bit | 名称 | Value | 说明 | |
| 47 | 起始位 | 0 |  | |
| 46 | 传输标志 | 0 | 表示本报文是响应报文，传输方向是SD卡🡪主机 | |
| 45:40 | 命令号 |  | 000000 | 本报文是对CMD0命令的响应 |
| 000001 | 本报文是对CMD1命令的响应 |
| …… | …… |
| 111111 | 本报文是对CMD63命令的响应 |
| 39:8 | 错误信息 |  | 在响应中使用32位数据标示SD卡在执行命令时遇到的错误和状态信息；  具体的错误信息见下表 | |
| 7:1 | CRC7校验 |  | CRC校验错误时，主机将忽视该响应 | |
| 0 | 终止位 | 1 |  | |
|  | | | |  | |
| R2响应  CID或CSD寄存器 | 135 | 起始位 | 0 |  | |
| 134 | 传输标志 | 0 |  | |
| 133:128 | 保留位 | 111111 |  | |
| 127:1 | CID/CSD |  | 如果命令是**CMD2**或**CMD10**，则响应CID寄存器内容  如果命令是**CMD9**，则响应CSD寄存器内容  CID和CSD寄存器中有CRC7校验，所以R2响应报文不使用CRC校验 | |
| 0 | 终止位 | 1 |  | |
|  | | | |  | |
| R3响应 OCR寄存器 | 47 | 起始位 | 0 |  | |
| 46 | 传输标志 | 0 |  | |
| 45:40 | 保留位 | 111111 |  | |
| 39:8 | OCR |  | OCR寄存器的值  OCR寄存器中有CRC7校验，所以R3响应报文不使用CRC校验 | |
| 7:1 | 保留位 | 1111111 |  | |
| 0 | 终止位 | 1 |  | |
|  | | | |  | |
| R6响应 RCA寄存器 | bit | 名称 | 说明 |  | |
| 47 | 起始位 | 0 |  | |
| 46 | 传输标志 | 0 |  | |
| 45:40 | 命令号 | 000011 | R6响应只对应命令CMD3 | |
| 39:8 | RCA |  | RCA寄存器的值 | |
| 7:1 | CRC7 |  |  | |
| 0 | 终止位 | 1 |  | |
|  | | | |  | |
| R7响应  RCA寄存器 | bit | 名称 | 说明 |  | |
| 47 | 起始位 | 0 |  | |
| 46 | 传输标志 | 0 |  | |
| 45:40 | 命令号 | 001000 | R6响应只对应命令CMD8 | |
| 39:20 | 保留位 | 00000 |  | |
| 19:16 | 接收电压 |  |  | |
| 15:8 | 检测模式 |  |  | |
| 7:1 | CRC7 |  |  | |
| 0 | 终止位 | 1 |  | |

## 常规数据包

如图所示，4线同步发送。

主机会发送命令告诉从机，发送的数据，一个数据包是多少个字节。

常规数据发送是先低字节，再高字节的顺序，但是每个字节则是先高位后低位



|  |  |
| --- | --- |
| 第1个CLK上升沿 | 4条数据线均发送一个起始位’0’ |
| 第2个CLK上升沿 | DAT3发送第1个字节的第7位，DAT2发送第1个字节的第6位，  DAT1发送第1个字节的第5位，DAT0发送第1个字节的第4位； |
| 在第3个CLK上升沿 | DAT3发送第1个字节的第3位，DAT2发送第1个字节的第2位，  DAT1发送第1个字节的第1位，DAT0发送第1个字节的第0位； |
| 在第4个CLK上升沿 | DAT3发送第2个字节的第7位，DAT2发送第2个字节的第6位，  DAT1发送第2个字节的第5位，DAT0发送第2个字节的第4位 |
| …… | …… |

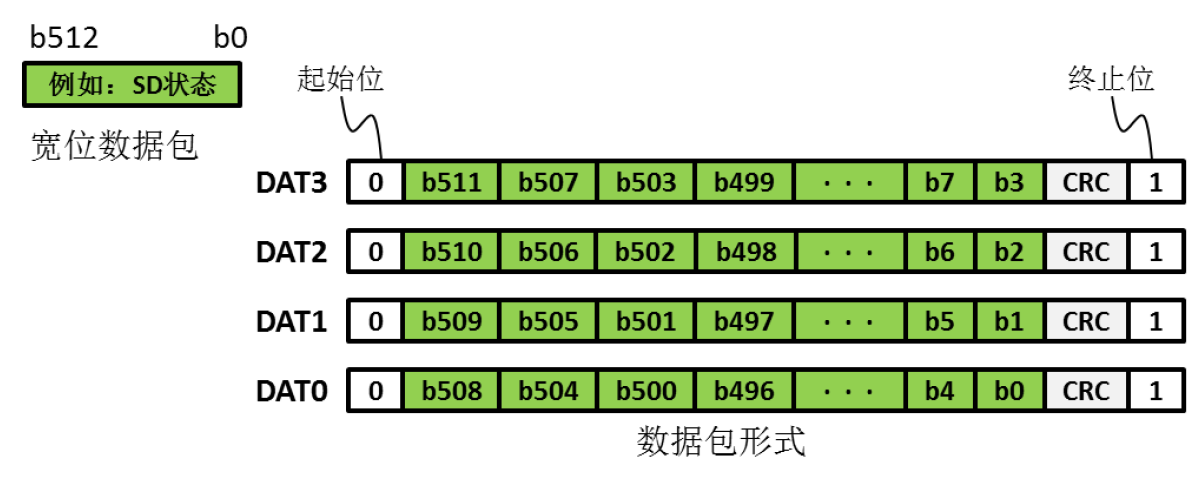
## 宽位数据包

对SD卡而言宽位数据包发送方式是针对SD卡SSR(SD状态)寄存器内容发送的，SSR寄存器总共有512bit，在主机发出ACMD13命令后SD卡将SSR寄存器内容通过DAT线发送给主机。

宽位数据从高位开始传输；DAT3数据线发较高位，DAT0数据线发较低位；数据每次传4Bit。

每条数据线上都要发送起始位、结束位以及CRC位。CRC 位每条数据线都要分别检查。

CRC 状态反馈和忙碌只是只会从DAT0 发送到host 端，DAT1-DAT3 忽略。



|  |  |
| --- | --- |
| 第1个CLK上升沿 | 4条数据线均发送一个起始位’0’ |
| 第2个CLK上升沿 | DAT3发送第511位，DAT2发送第510位，DAT1发送第509位，DAT0发送第508位； |
| 第3个CLK上升沿 | DAT3发送第507位，DAT2发送第506位，DAT1发送第505位，DAT0发送第504位； |
| …… | …… |

## 状态数据块

512 位的扩展状态信息，支持SD 卡的特定功能以及将来的应用特定功能。

SD 状态包含了与sd 卡属性功能相关的状态位，可能会用于将来特定的应用命令。

SD状态的大小是一个512bit 的数据块。这个寄存器的内容会通过DAT 总线传递给主机，CRC16。

SD 状态会通过DAT 总线发送给主机，作为ACMD13(前面是CMD55)的响应。ACMD13 只能在“transfer”模式发送给已选定的卡。卡状态结构体在下表描述。“类型”和“清除条件”使用的缩写和卡状态一样。

|  |  |  |
| --- | --- | --- |
| SD状态 | | |
| 位 | 标识 | 含义 |
| 511:510 | DAT\_BUS\_WIDTH | SET\_BUS\_WIDTH 定义的当前总线宽度 |
| 509 | SECURED\_MODE |  |
| 508:496 |  |  |
| 495:480 | SD\_CARD\_TYPE | 在将来，8 位LBS 将会被用来定义SD卡的不同的变化(每一位会定义不同的SD 类型)。8 位MSB 会被用来定义不遵循当前Sd 物理协议的SD 卡 |
| 479:448 | SIZE\_OF\_PROTECTED\_AREA | 这个设置在标准和高容量卡之间是有不同的。  标准卡，保护区域的容量是这么计算的：  保护区域= SIZE\_OF\_PROTECTED\_AREA \* MULT\*BLOCK\_LEN  SIZE\_OF\_PROTECTED\_AREA 是以MULT\*BLOCK\_LEN 为单位。  高容量卡，保护区域的容量是这么计算的：  保护区域= SIZE\_OF\_PROTECTED\_AREA  SIZE\_OF\_PROTECTED\_AREA 以byte 为单位 |
| 447:440 | SPEED\_CLASS | 8bit 的数据，表示速度等级，数值是2 的整数倍  0000 0000： Class0  0000 0001： Class2  0000 0010: Class4  0000 0011: Class6 |
| 439:432 | PERFORMANCE\_MOVE | 8 位区域表明性能，以及可以设置的值[MB/s]。  如果卡不使用RU 搬迁数据，那么Pm 是无穷大的。  0000 0000： 1Mb/s  0000 0001： 2Mb/s  0000 0010: 3Mb/s  0000 0011: 4Mb/s  ……  1111 1110： 254Mb/s  1111 1111：无穷大 |
| 431:428 | AU\_SIZE | 4bit 数值，AU 的长度，值是16\*(2^n)  0000： 16Kb  0001： 32Kb  0010： 64Kb  0011： 128Kb  0100： 256Kb  0101： 512Kb  0110： 1Mb  0111： 2Mb  1000： 4Mb  最大的AU 长度，取决于卡的容量。卡可以设置任意AU 长度，要介于RU  长度和最大AU 长度之间。  容量 16MB-64MB 128MB-256MB 512MB 1GB-32GB  最大AU 长度 512KB 1MB 2MB 4MB  注意：主机应该使用最大AU 长度(4MB)来决定主机的缓存大小。主机可以协调多个AU 使用同一个单元。 |
| 427:424 |  |  |
| 423:408 | ERASE\_SIZE SR | 16bit 长度，代表N.erase。当N.erase 个AU 被擦除后，超时值由ERASE\_TIMEOUT 指定。主机应该决定一次操作中要擦除的AU 数，这样主机就能够表明擦除操作的进程。如果值是0，那么不支持擦除超时的计算 |
| 407:402 | ERASE\_TIMEOUT | 6bit 长度，表明T.erase，值表明了当ERASE\_SIZE 定义的多AUs 被擦除时，偏移的擦除超时。ERASE\_TIMEOUT 的范围被定义到63 秒，而且卡厂家可以根据情况改变ERASE\_SIZE 和ERASE\_TIMEOUT 的组合。一旦ERASE\_TIMEOUT 确定了，那么ERASE\_SIZE 也确定了。主机可以通过方程式(6)确定任意数目的擦除超时。参考4.14 中的擦除超时计算方法。如果ERASE\_SIZE 被设置为0，这个值也是0。 |
| 401:400 | ERASE\_OFFSET | 2bit 数据，表明T.offset，有4 种选择。擦除偏移通过上位平移来调整线。参考表4-33  以及4.14 的公式(6)。  如果ERASE\_SIZE 和ERASE\_TIMEOUT 是0 的话，这个值无意义。 |
| 399:312 |  |  |
| 311:0 |  |  |

# SDIO命令

## 概述

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class1基础  命令 | Index | 名称 | 类型 | Response | Argument | | |
| CMD0 | GO\_IDLE\_STATE | 无响应广播命令 | / | 填充位 | | |
| CMD2 | ALL\_SEND\_CID | 广播命令 | R2 | 填充位 | | |
| CMD3 | SEND\_RELATIVE\_ADDR | 广播命令 | R6 | 填充位 | | |
| CMD4 | SET\_DSR | 无响应广播命令 | / | 31:16 | DSR | |
| 15:0 | 填充位 | |
| CMD7 | SELECT/  DESELECT\_CARD | 寻址命令 | R1b | 31:16 | RCA | |
|  | 15:0 | 填充位 | |
| CMD8 | SEND\_IF\_COND | 无响应广播命令 | R7 | 31:12 | 保留位=0 | |
| / | 11:8 | VHS---主机指定测试的从机支持电压  0000 ：未定义  0001 ：2.7~~3.6V  0010 ：预留给低电卡  0100 ：保留  1000 ：保留 | |
| / | 7:0 | 检查模式，  任意的8bit 模式(推荐使用10101010b)  主机会通过CRC 和检查模式来确认通信的有效性 | |
| CMD9 | SEND\_CSD | 寻址命令 | R2 | 31:16 | RCA | |
| 15:0 | 填充位 | |
| CMD10 | SEND\_CID | 寻址命令 | R2 | 31:16 | RCA | |
| 15:0 | 填充位 | |
| CMD12 | STOP\_TRANSMISSION | 寻址命令 | R1 | 填充位 | | |
| CMD13 | SEND\_STATUS | 寻址命令 | R1 | 31:16 | RCA | |
| 15:0 | 填充位 | |
| CMD15 | GO\_INACTIVE\_STATE | 寻址命令 | / | 31:16 | RCA | |
| 15:0 | 填充位 | |
| Class 2  面向块  读操作 | CMD16 | SET\_BLOCKLEN | 寻址命令 | 无 | 31:0 | 命令块长度 | |
| CMD17 | READ\_SINGLE\_BLOCK | 寻址数据  传输命令 | R1 | 31:0 | 数据地址 | |
| CMD18 | READ\_MULTIPLE\_BLOCK | 寻址数据  传输命令 | R1 | 31:0 | 数据地址 | |
| Class 4面向块  写操作 | CMD16 | SET\_BLOCKLEN | 寻址命令 | 无 | 31:0 | 命令块长度 | |
| CMD24 | WRITE\_BLOCK | 寻址数据  传输命令 | R1 | 31:0 | 数据地址 | |
| CMD25 | WRITE\_MILTIPLE\_BLOCKK | 寻址数据  传输命令 | R1 | 31:0 | 数据地址 | |
| CMD27 | PROGRAM\_CSD | 寻址数据  传输命令 | R1 | 31:0 | 填充位 | |
| Class 6  写保护命令 | CMD28 | ERASE\_WR\_BLK\_START | 寻址命令 | R1 | 31:0 | 数据地址 | 擦除的起始块地址 |
| CMD29 | ERASE\_WR\_BLK\_END | 寻址命令 | R1 | 31:0 | 数据地址 | 擦除的结束块地址 |
| CMD30 |  | 寻址命令 | R1b | 31:0 | 数据地址 | 擦除预先选定的块 |
| Class7  加锁  命令 | CMD16 | SET\_BLOCKLEN | 寻址命令 | 无 | 31:0 | 命令块长度 | |
| CMD42 | LOCK\_UNLOCK | 寻址数据  传输命令 | R1 | 31:0 | 保留 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class8  特定  应用  命令 | CMD55 | APP\_CMD | 寻址命令 | R1 | 31:16 | RCA |
|  |  |  |  | 15:0 | 填充位 |
| CMD56 | GEN\_CMD | 寻址数据传输命令 | R1 | 31:1 | 由厂家定义数据块的的格式和含义  卡应该预防接收到未知格式的情况。 |
| 0 | 传输方向  0：写  1：读 |
| ACMD6 | SET\_BUS\_WIDTH | 寻址数据传输命令 | R1 | 31:2 | 填充位 |
| 1:0 | 总线宽度：  00 表示1bit，10 表示4bit。  接受的数据总线定义在SCR 寄存器中。 |
| ACMD13 | SD\_STATUS | 寻址数据传输命令 | R1 | 31:0 | 填充位 |
| ACMD22 | SEND\_NUM\_WR\_BLOCK | 寻址数据传输命令 | R1 | 31:0 | 填充位 |
| ACMD23 | SET\_WR\_BLK\_ERASE\_COUNT | 寻址命令 | R1 | 31:0 | 填充位 |
| ACMD41 | SD\_SEND\_OP\_COND | 带响应广播命令 | R3 | 31 | 保留位==0 |
| 30 | HCS |
| 29:24 | 保留位=0 |
| 23:0 | OCR  VDD电压 |
| ACMD42 | SET\_CLR\_CARD\_DETECT | 寻址命令 | R1 | 31:1 | 填充位 |
| 0 | 设置卡检测  1-connect，0-disconect |
| ACMD51 | SEND\_SCR | 寻址数据传输命令 | R1 | 31:0 | 填充位 |
| Class9  I/O模式  命令 | CMD52 |  |  |  |  |  |
| CMD53 |  |  |  |  |  |
| CMD54 |  |  |  |  |  |
| Class 10切换  命令 | CMD6 |  |  |  | 31 | 模式  0:查询  1:切换 |
| 30:24 | 保留，全0 |
| 23:20 | 保留给组6(0h 或Fh) |
| 19:16 | 保留给组5 |
| 15:12 | 保留给组4 |
| 11:8 | 保留给组3 |
| 7:4 | 功能组2  -命令系统 |
| 3:0 | 功能组1  -访问模式 |
| CMD34 |  |  |  |  |  |
| CMD35 |  |  |  |  |  |
| CMD36 |  |  |  |  |  |
| CMD37 |  |  |  |  |  |
| CMD50 |  |  |  |  |  |
| CMD57 |  |  |  |  |  |

## 基本命令(Class 0)

|  |  |
| --- | --- |
| Index | 说明 |
| CMD0 | 复位所有的卡到idle(空闲)状态。 |
| CMD2 | 通知所有卡通过CMD线返回各自的CID值。 |
| CMD3 | 通知所有卡发布新RCA |
| CMD4 | 编程所有卡的DSR寄存器 |
| CMD7 | 选择/取消 选择RCA地址卡。  这个命令在“stand-by”和“transfer”状态之间使用，以及“programming”和“disconnect”状态之间使用。  在这两种情况下，地址匹配就被选中，不匹配就被取消选中。  address 0，取消所有卡的选中。  当RCA=0 时，主机可能做下面的事情：  -使用其他的RCA 号来取消卡  -重新发送CMD3 来改变卡的RCA 号，使它不为0。然后在用RCA=0 来取消选择。 |
| CMD8 | 电压检测：  CMD8 要在SD 卡处于“idle”状态下使用  主机不知道SD卡接口支持的电压的情况下可以使用这个命令验证SD 卡接口操作条件。  SD卡收到命令后分析VHS参数。如果卡能够在该电压下操作，SD卡发送R7响应至主机  响应参数中卡反馈了电压范围和检测模式设置。  如果卡不能在支持电压下操作，SD卡就不会发送响应，并保持在“idle”状态。  使能已存在命令的扩展和响应：  CMD8 要在SD 卡处于“idle”状态下使用  恢复CMD8 就能够通过从定义之前预留的bit，为一些已存在的命令扩展新的功能。  如ACMD41 就是被扩展用来支持高容量卡的。  当卡在“idle”状态时，主机应该先发送CMD8，再发送ACMD41。 |
| CMD9 | 选定卡通过CMD线发送CSD内容 |
| CMD10 | 选定卡通过CMD线发送CID内容 |
| CMD12 | 强制卡停止传输 |
| CMD13 | 选定卡通过CMD线发送它状态寄存器 |
| CMD15 | 使选定卡进入“inactive”状态  这个命令用于当主机明确的想停用这张卡的时候。保留位要设置为‘0’ |

## 

## 面向块的读操作(Class 2)

|  |  |
| --- | --- |
| Index | 说明 |
| CMD16 | 对于标准SD 卡来说，这个命令会设置所有数据块的长度(字节)。  默认的块长度是512Byte。  只有当CSD 允许部分块读取操作，设置的长度才对存储访问命令有效。  对于高容量SD 卡，数据块长度是固定为512Byte，CMD16 无效，此时这个命令只对加锁/解锁命令有效。  不管是哪种SD卡，只要数据块长度设置大于512Byte，就报错BLOCK\_LEN\_ERROR。  当以块地址模式进行存储读和写命令时，不管CMD16 设置的块长是多少，块的长度都固定为512Byte。就是说块长度的设置不会影响存储访问命令。  CMD42(锁定/解锁)不属于存储访问命令，因此块长度可以通过CMD16 进行设置。不管卡容量多少，只要块长  度设置大于512Byte，都会报错BLOCK\_LEN\_ERROR。 |
| CMD17 | 如果是标准卡，使用该命令可以 读取SEL\_BLOCK\_LEN长度字节的数据块；  对于SDHC卡，使用该命令可以 读取512字节的数据块。 |
| CMD18 | 连续从SD卡读取数据块，直到被CMD12中断。  传输的数据块长度同CMD17 |

注：

①:数据传输不应该穿过物理块的边界，除非CSD 中设置了READ\_BLK\_MISALIGH。如果部分块写操作不支持，那么块长度=默认块长度(CSD 中定义了)。

②:数据地址在标准卡中是以字节为单位的，而高容量卡中，是以块(512byte)为单位的。

## 面向块的写操作(Class 4)

①:32bit 的写保护位会通过数据线发送，后面跟着16CRC。写保护位的最后1bit(最小位)对应着第1 个地址组。如果最后一个组的地址超过了有效范围，那么对应的写保护位就会设置为1。

②:标准卡的数据地址以字节为单位。

|  |  |
| --- | --- |
| Index | 说明 |
| CMD16 | 对于标准SD 卡来说，这个命令会设置所有数据块的长度(字节)。  默认的块长度是512Byte。  只有当CSD 允许部分块读取操作，设置的长度才对存储访问命令有效。  对于高容量SD 卡，数据块长度是固定为512Byte，CMD16 无效，此时这个命令只对加锁/解锁命令有效。  不管是哪种SD卡，只要数据块长度设置大于512Byte，就报错BLOCK\_LEN\_ERROR。 |
| CMD24 | 对于标准卡，写入SEL\_BLOCK\_LEN长度字节的块；  对于SDHC卡，写入512字节的块。 |
| CMD25 | 连续向SD卡写入数据块，直到被CMD12中断。每块长度同CMD17。 |
| CMD27 | 对CSD的可编程位进行编程 |

## 擦除命令(Class 5)

同时擦除很多数据块有助于提高数据的吞吐量。

这些写数据块的标识是通过CMD32--ERASE\_WR\_BLK\_START，CMD33---ERASE\_WR\_BLK\_END命令来完成的。

主机应该遵循下面的命令序列：CMD32 - > CMD33 -> CMD38(ERASE)。如果顺序不对，卡就会设置ERASE\_SEQ\_ERROR 位到状态寄存器，并且重启整个序列。

如果收到序列外的命令(除了SEND\_STATUS)，卡会设置ERASE\_RESET 状态位到寄存器中，重启擦除序列，并且执行最后那个命令。

如果擦除范围包含了写保护的部分，写保护的部分不会动，而只擦除非保护的部分，并且状态寄存器中的WP\_ERASE\_SKIP 位应该设置。

地址设置命令里设置的地址区域是写块区域，以Byte 作为单位。

如块写部分的描述一样，卡会DAT0 拉低来表明正在进行擦除操作。真正的擦除时间可能会很长，主机可能发送CMD7 来取消卡的选定或者使卡进入disconnect 状态。

擦除后的卡上的数据可能是‘ 0 ’ 或者‘ 1 ’， 取决于卡的厂家。SCR 寄存器的DATA\_STAT\_AFTER\_ERASE(bit55)定义了它是‘0’还是‘1’。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | 名称 | 类型 | Response | Argument | | 说明 |
| CMD32 | ERASE\_WR\_BLK\_START | 寻址命令 | R1 | 31:0 | 数据地址 | 设置擦除的起始块地址 |
| CMD33 | ERASE\_WR\_BLK\_END | 寻址命令 | R1 | 31:0 | 数据地址 | 设置擦除的结束块地址 |
| CMD38 |  | 寻址命令 | R1b | 31:0 | 数据地址 | 擦除预先选定的块 |

①:标准卡是字节为单位，高容量卡是块为单位(512Byte)

## 写保护命令(Class 6)

|  |  |
| --- | --- |
|  |  |
| 物理写保护开关(主机支持) | 卡的侧面有一个物理滑片(见第8 章)被用来标记卡是否被写保护了。 |
| 卡内部写保护卡支持) | 卡内部写保护(可选) |
| 密码保护卡锁定操作 | 卡的数据可以被写保护，以免被擦除或者覆盖。整个卡可能被厂家或者内容提供者，通过设置CSD 的永久或临时写保护位， 而永久的写保护。对于支持通过设置CSD 的WP\_GRP\_ENABLE 位来实现数据写保护的卡来说，数据的一部分可以被保护(单位是CSD 中规定的WP\_GRP\_SIZE 扇区)，而且写保护可以被用户改变。SET\_WRITE\_PROT 命令设置写保护区域的写保护，而CLR\_WRITE\_PROT 命令会清楚写保护区域的写保护。 |

SEND\_WRITE\_PROT 命令同单块读命令一类似。卡会发送一个数据块在16 位CRC 位之后有32 位写保护位(表示了以指定的地址开始的32 个写保护组)。写保护命令中的地址是一组以byte 为单位的地址。卡会忽略所有LSB 的低于组的部分。

密码卡锁保护在后面的章节中介绍

注意：高容量SD 卡不支持写保护，发送CMD28，CMD29，CMD30 命令给高容量卡会会报错ILLEGAL\_COMMAND

## 加锁命令(Class 7)

SD 卡支持3 中写保护方法：

—物理写保护开关(主机支持)

—卡内部写保护(卡支持)

—密码保护卡锁定操作

物理写保护开关

卡的侧面有一个物理滑片(见第8 章)被用来标记卡是否被写保护了。

卡内部写保护(可选)

卡的数据可以被写保护，以免被擦除或者覆盖。整个卡可能被厂家或者内容提供者，通过设置CSD 的永久或临时写保护位， 而永久的写保护。对于支持通过设置CSD 的WP\_GRP\_ENABLE 位来实现数据写保护的卡来说，数据的一部分可以被保护(单位是CSD 中规定的WP\_GRP\_SIZE 扇区)，而且写保护可以被用户改变。SET\_WRITE\_PROT 命令设置写保护区域的写保护，而CLR\_WRITE\_PROT 命令会清楚写保护区域的写保护。

SEND\_WRITE\_PROT 命令同单块读命令一类似。卡会发送一个数据块在16 位CRC 位之后有32 位写保护位(表示了以指定的地址开始的32 个写保护组)。写保护命令中的地址是一组以byte 为单位的地址。卡会忽略所有LSB 的低于组的部分。

密码卡锁保护在后面的章节中介绍

注意：高容量SD 卡不支持写保护，也不会响应写保护命令(CMD28,CMD29,CMD30)

|  |  |
| --- | --- |
| Index | 说明 |
| CMD16 | 对于标准SD 卡来说，这个命令会设置所有块命令的长度(字节)。  默认的块长度是512Byte。  只有当CSD 允许部分块读取操作，设置的长度才对存储访问命令有效。  对于高容量SD 卡来说，CMD16 设置的块长度对于读写命令来说没有硬性，因为块长度是固定的512Byte。  这个命令只对加锁/解锁命令有效。  不管哪种，只要块长度设置大于512Byte，就报错BLOCK\_LEN\_ERROR。 |
| CMD42 | 加锁/解锁SD卡  用来设置/复位密码，或者加锁/解锁卡。  数据块的长度是通过SET\_BLOCK\_LEN 命令设置的。  参数以及锁卡数据结构里的保留位应该设置为0 |

## 特定应用命令(Class 8)

|  |  |
| --- | --- |
| Index | 说明 |
| CMD55 | 指定下个命令为特定应用命令，不是标准命令  主机要发送SD卡制造商特定的ACMD命令如ACMD6，需要在发送该命令之前无发送CMD55命令，告知SD卡接下来的命令为特定应用命令。**CMD55命令只对紧接的第一个命令有效**  从机SD卡如果检测到CMD55命令，就会返回R1响应至主机，响应里有APP\_CMD 为设置信号，表明是作为ACMD 解析的，告诉主机现在已经准备好接受ACMD 命令了。  然后对接下来收到的的第一条命令进行使用非常规操作，执行其特定应用功能。  如果这个“接下来的第一条命令”是非ACMD命令，那么从机就会把他作为普通的sd 卡命令响应，从机寄存器中的APP\_CMD 位会被清除。  注意：一般主机都会在发送CMD55后紧接着发送一个ACMD命令，但如果没有主机没有这么做，即主机发送了CMD55后又发送了一个常规CMD命令，那么SD卡对收到这个常规CMD命令仍会按标准命令来执行。  如果多个CMD55 被连续发送，那么每个响应的APP\_CMD 位都会被设置为1。最后一个CMD55 后面紧跟的命令会被当作ACMD 命令。如果超过一个命令跟在CMD55 后(CMD55 除外)，只有第一个命令被当作ACMD 命令。后面的都作为常规命令。  如果发送了一个无效命令，就会返回常规的命令错误。  从SD 卡的协议来看，ACMD 号应该由厂家参照某些限制来定义。下面的ACMD 是保留的，任何厂家都不应该使用它们：ACMD6,ACMD13，ACMD17-25，ACMD38-49，ACMD51。 |
| CMD56 | 通用命令，或者特定应用命令中，用于传输一个数据块，通用卡数据块长度由SET\_BLOCK\_LEN 命令设置  最低位为1表示读数据，为0表示写数据  GEN\_CMD(CMD56)厂家特定以及可选的命令。本文档定义了这个命令。总线上的传输，  GEN\_CMD 和“单块读/写”(CMD24、CMD7)命令是一样的，也只能在“transfer”状态使用。  不同之处在于，参数指示了数据传输的方向(而不是地址)，并且数据块不是一个有效的存储数据，但是它有厂家特定的。  发送CMD56 之前，需要让卡进入“transfer”状态。如果是标准SD 卡，数据块的长度是CMD16 中定义的BLOCK\_LEN。如果是高容量卡，数据块的大小固定为512Byte。  厂家可以定义特定的格式到参数的bit31-1，以及这个命令的数据块内容。然而，我们需要考虑到，在使用CMD56 之前，主机应该确认CID 信息，这样卡才能支持CMD56 的格式。 |
| ACMD6 | 定义数据总线宽度  宽总线(4Bit 宽)操作模式可以通过命令ACMD6 来选定和取消。上电或者GO\_IDLE(CMD0)命令后，默认的总线宽度是1Bit；  如果想要改变总线宽度，需要具备下面两个条件:  a) 卡处于transfer 状态  b) 卡没有被锁定(锁定的卡会认为ACMD6 是无效命令)  SD卡收到CMD32命令后返回R1响应至主机。 |
| ACMD13 | 发送SD状态  SD卡收到CMD32命令后返回R1响应至主机。 |
| ACMD22 | 发送已经写入的块(没有错误的)的数目。  响应32bit + CRC 数据块。  如果WRITE\_BL\_PARTIAL=0，ACMD22 的单位总是512Byte。  如果WRITE\_BL\_PARTIAL=1，ACMD22 的单位就是写命令使用的单块的长度。  SD卡收到CMD32命令后返回R1响应至主机。 |
| ACMD23 | 设置需要擦除的块的数目(提高速度)。  默认值=1。  STOP\_TRAN(CMD12)用来停止多块的数据传输，不管ACMD23 功能是否使用。  SD卡收到CMD32命令后返回R1响应至主机。 |
| ACMD41 | 主机要求卡发送它的支持信息  (HCS)OCR寄存器内容。  并要求卡通过命令线返回OCR 寄存器内容。  当卡收到SEND\_IF\_COND 时，HCS 是有效的。  SD卡收到ACMD41命令后返回R3响应至主机。 |
| ACMD42 | 50k 欧姆上拉电阻  DAT3/CD  SD卡收到ACMD42命令后返回R1响应至主机。 |
| ACMD51 | 读取配置寄存器SCR  SD卡收到ACMD42命令后返回R1响应至主机。 |

## I/O模式命令(Class 9)

## 切换命令(Class 10)

## SDIO响应

R1响应的错误信息表

|  |  |  |
| --- | --- | --- |
|  | | |
| bit | 标识 | 含义 |
| 31 | OUT\_OF\_RANGE | 命令的参数超出卡的接受范围 |
| 30 | ADDRESS\_ERROR | 没对齐的地址，同命令中使用的块长度不匹配 |
| 29 | BLOCK\_LEN\_ERROR | 卡不接受的块长度，或者传输的长度同块长度不匹配 |
| 28 | ERASE\_SEQ\_ERROR | 擦除命令序列有错 |
| 27 | ERASE\_PARAM | 要擦除的“写模块”的选择无效 |
| 26 | WP\_VIOLATION | 0：没有保护 1：对保护模块进行写操作；对写保护卡进行操作 |
| 25 | CARD\_IS\_LOCKED | SD卡被主机加锁 |
| 24 | LOCK\_UNLOCK\_FAILED | 加锁解锁命令错误 |
| 23 | COM\_CRC\_ERROR | 前一个命令的CRC 检查错误 |
| 22 | ILLEGAL\_COMMAND | 前一个命令是卡当前状态不能接受的命令 |
| 21 | CARD\_ECC\_FAILED | 内部ECC 收到但是数据不对 |
| 20 | CC\_ERROR | 内部的卡控制器错误 |
| 19 | ERROR | 通用或者未知错误 |
| 18：17 | 保留位=0 | |
| 16 | CSD\_OVERWRITE | CSD 的只读部分同卡的内容不匹配；试图方向拷贝，或者永久写保护位 |
| 15 | WP\_ERASE\_SKIP | 因为保护块，或者是写保护卡，只擦除了部分指定的地址。 |
| 14 | CARD\_ECC\_DISABLED | 0：命令已经执行，使用内部ECC  1：命令已经执行，没使用内部ECC |
| 13 | ERASE\_RESET | 0：收到退出擦除序列命令，在执行之前，擦除序列没有被清除  1：因为收到退出擦除序列命令，在执行之前，擦除序列被清除 |
| 12:9 | CURRENT\_STATE | SD卡当前状态  0000：  0001：  0010：  0011：  0100：  0101：  0110：  0111： |
| 8 | READY\_FOR\_DATA | 0：缓存非空，不可以接收数据  1：总线上表示缓存空的状态，即可以接受数据 |
| 7：6 | 保留位=0 | |
| 5 | APP\_CMD | 0：卡不会接收到ACMD 命令，命令没有被作为ACMD 解读  1：卡会接收到ACMD 命令。或者说命令已经被作为ACMD 解读 |
| 4 | 保留位=0 | |
| 3 | AKE\_SEQ\_ERROR | 认证过程序列发生错误 |
| 2：0 | 保留位=0 | |

## CAN

|  |  |  |  |
| --- | --- | --- | --- |
| 标准  数据帧 | Start | 1 bit | Start=0  用于通知各个节点将有数据传输，其它节点通过帧起始信号的电平跳变沿来进行硬同步。  只在总线空闲时，才允许站开始发送（信号）。所有的站必须同步于首先开始发送信息的站的帧起始前沿 |
| CAN-ID | 11bit | 功能码(4bit)+Node-ID(7bit) |
| RTR | 1 | RTR=0表示标准数据帧 |
| IDE | 1 bit | IDE=0，表示标准数据帧 |
| r0 | 1 bit | 保留位，r0=0 |
| DLC | 4 bits | 数据长度，范围0000~1000，单位B |
| Data | 1~8B | 数据内容，整数字节 |
| CRC Sequence | 15 bits | 为进行CRC计算，被除的多项式系数由无填充位流给定，组成这些位流的成分是：帧起始、仲裁场、控制场、数据场（假如有），而15 个最低位的系数是0。  将此多项式被下面的多项式发生器除（其系数以2 为模）：  X 15 + X 14 + X 10 + X 8 + X 7 + X 4 + X 3 + 1  这个多项式除法的余数就是发送到总线上的CRC 序列（CRC SEQUENCE）。  为了实现这个功能，可以使用15 位的位移寄存器CRC\_RG（14:0）。如果用NXTBIT 标记指示位流的下一位，它由从帧的起始到数据场末尾都由无填充的位序列给定。  CRC 序列（CRC SEQUENCE）的计算如下：  CRC\_RG = 0; // 初始化移位寄存器  REPEAT;  CRCNXT = NXTBIT EXOR CRC\_RG（14）;  CRC\_RG（14:1） = CRC\_RG（13:0）; // 寄存器左移1 位  CRC\_RG（0） = 0;  IF CRCNXT THEN  CRC\_RG（14:0） = CRC\_RG（14:0） EXOR （4599hex）;  ENDIF  UNTIL （CRC 序列开始或存在一个错误条件）  **先发送CRC SEQUENCE再发送CRC DELIMITER**  **CRC SEQUENCE中先发送高位** |
| CRC Delimiter | 1bit | 用于隔开CRC Sequence和ACK Slot |
| ACK Slot | 1 bit | 发送端发送报文时，ACK Slot=1  接收端接收到该报文后，回复该报文时，ACK SLOT=0  **先发送ACK Slot再发送ACK Delimiter** |
| ACK DELIMITER | 1 bit | ACK Delimiter=1，用于隔开ACK Slot 和End of Frame；，ACK Slot被两个“隐性”位包围 |
| EOF | 7 bits | 帧结束 ，EOF=111 1111 |
| 扩展  数据帧 | Start | 1 bit | Start=0 |
| CAN-ID | 11bit | 功能码(4bit)+Node-ID(7bit) |
| IDE | 1 bit | IDE=1，表示扩展数据帧 |
| SRR | 1 bit | SRR=1 |
| Extended ID | 18bits |  |
| 保留位 | 2bit | 00 |
| DLC | 4 bits | 数据长度，范围0000~1000，单位B |
| Data | 1~8B | 数据内容，整数字节 |
| CRC Sequence | 15 bits |  |
| ACK Slot | 1 bit | 发送端发送报文时，ACK Slot=1  接收端接收到该报文后，回复该报文时，ACK SLOT=0 |
| ACK DELIMITER | 1 bit | ACK Delimiter=1，用于隔开ACK Slot 和End of Frame；，ACK Slot被两个“隐性”位包围 |
| EOF | 7 bits | 帧结束 ，EOF=111 1111 |

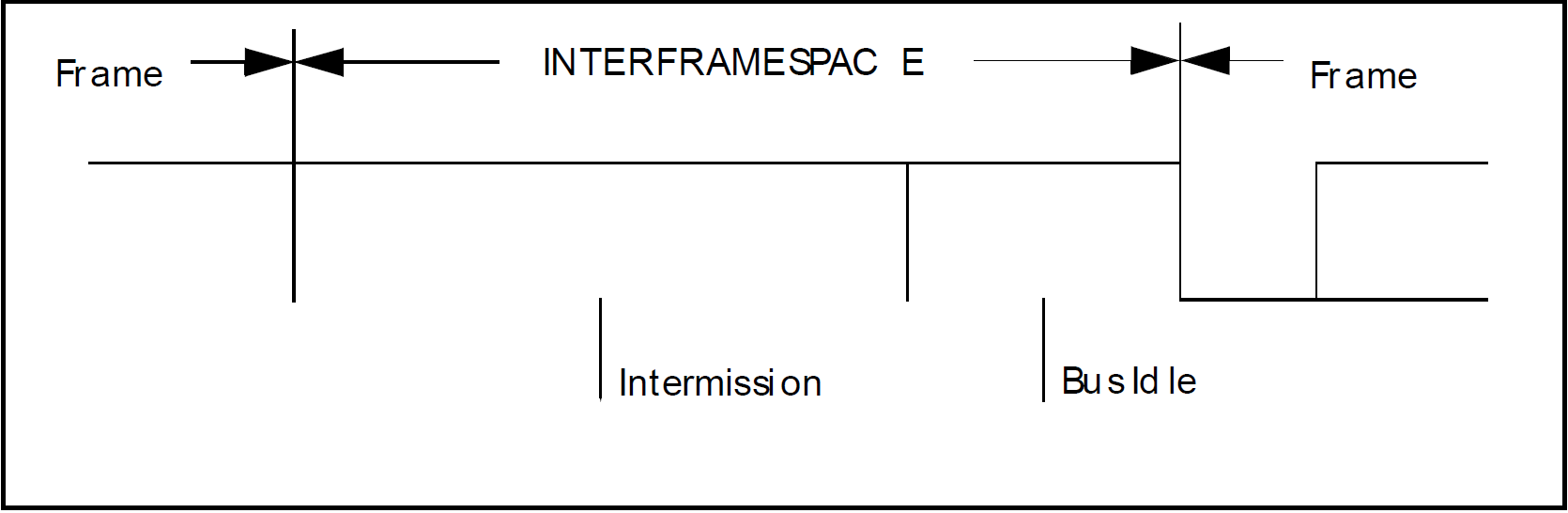
|  |  |  |  |
| --- | --- | --- | --- |
| 远程帧 | Start | 1 bit | Start=0 |
| CAN-ID | 11bit | 功能码(4bit)+Node-ID(7bit) |
| RTR | 1 | RTR=1，表示远程帧 |
| IDE | 1 bit | IDE=0，表示标准数据帧 |
| 保留位 | 1 bit | 0 |
| DLC | 4 bits | 数据长度，范围0000~1000，单位B |
| CRC Sequence | 15 bits |  |
| CRC Delimiter | 1bit | 用于隔开CRC Sequence和ACK Slot |
| ACK Slot | 1 bit | ACK Slot=1 |
| ACK DELIMITER | 1 bit | ACK Delimiter=1，用于隔开ACK Slot 和End of Frame；，ACK Slot被两个“隐性”位包围 |
| EOF | 7 bits | 帧结束，EOF=111 1111 |
| 错误帧 | Error Flag | 6 bits | 有两种形式的错误标志，主动错误标志（Active error flag）和被动错误标志（Passive error flag）。  主动错误标志由6 个连续的“显性”位组成。  被动错误标志由6 个连续的“隐性”的位组成，除非被其他节点的“显性”位重写。  为了能正确地终止错误帧，一“错误被动”的节点要求总线至少有长度为3个位时间的总线空闲（如果“错误被动”的接收器有本地错误的话）。因此，总线的载荷不应为100%。  检测到错误条件的“错误主动”的站通过发送主动错误标志，以指示错误。错误标志的形式破坏了从帧起始到CRC 界定符的位填充规则（参见“编码”），或者破坏了应答场或帧末尾场的固定形式。所有其他的站由此检测到错误条件并与此同时开始发送错误标志。  因此，“显性”位（此“显性”位可以在总线上监视）的序列导致一个结果，这个结果就是把各个单独站发送的不同的错误标志叠加在一起。这个顺序的总长度最小为6 个位，最大为12 个位。  检测到错误条件的“错误被动”的站试图通过发送被动错误标志，以指示错误。“错误被动”的站等待6 个相同极性的连续位（这6 个位处于被动错误标志的开始）。当这6个相同的位被检测到时，被动错误标志的发送就完成了。 |
| Error Flag | 6 bits |  |
| …… | …… | …… |
| Error Delimiter | 8 bits | Error Delimiter = 1111 1111  错误标志传送了以后，每一站就发送“隐性”的位并一直监视总线直到检测出一个“隐性”的位为止。  然后就开始发送7 位以上的“隐性”位。 |
| 过载帧 | OverLoad Flag | 6 bits | 过载标志由6 个“显性”的位组成。过载标志的所有形式和主动错误标志的一样。  过载标志的形式破坏了间歇场的固定形式。因此，所有其他的站都检测到一过载条件并与此同时发出过载标志。（万一有的节点在间歇的第3 个位期间于本地检测到“显性”位，则其他的节点将不能正确地解释过载标志，而是将这6 个“显性”位中的第一个位解释为帧的起始。这第6 个“显性”的位破坏了产生错误条件的位填充的规则。 |
| OverLoad Flag | 6 bits |  |
| …… | …… | …… |
| OverLoad Delimiter | 8 bits | OverLoad Delimiter = 1111 1111  过载界定符包括8 个“隐性”的位。  过载界定符的形式和错误界定符的形式一样。过载标志被传送后，站就一直监视总线直到检测到一个  从“显性”位到“隐性”位的发送（过渡形式）。此时，总线上的每一个站完成了过载标志的发送，并开  始同时发送7 个以上的“隐性”位。 |

|  |
| --- |
| 有两种过载条件都会导致过载标志的传送：  1. 接收器的内部条件（此接收器对于下一数据帧或远程帧需要有一延时）。  2. 间歇场期间检测到一“显性”位。  由过载条件1 而引发的过载帧只允许起始于所期望的间歇场的第一个位时间开始。而由过载条件2 引  发的过载帧应起始于所检测到“显性”位之后的位。 |

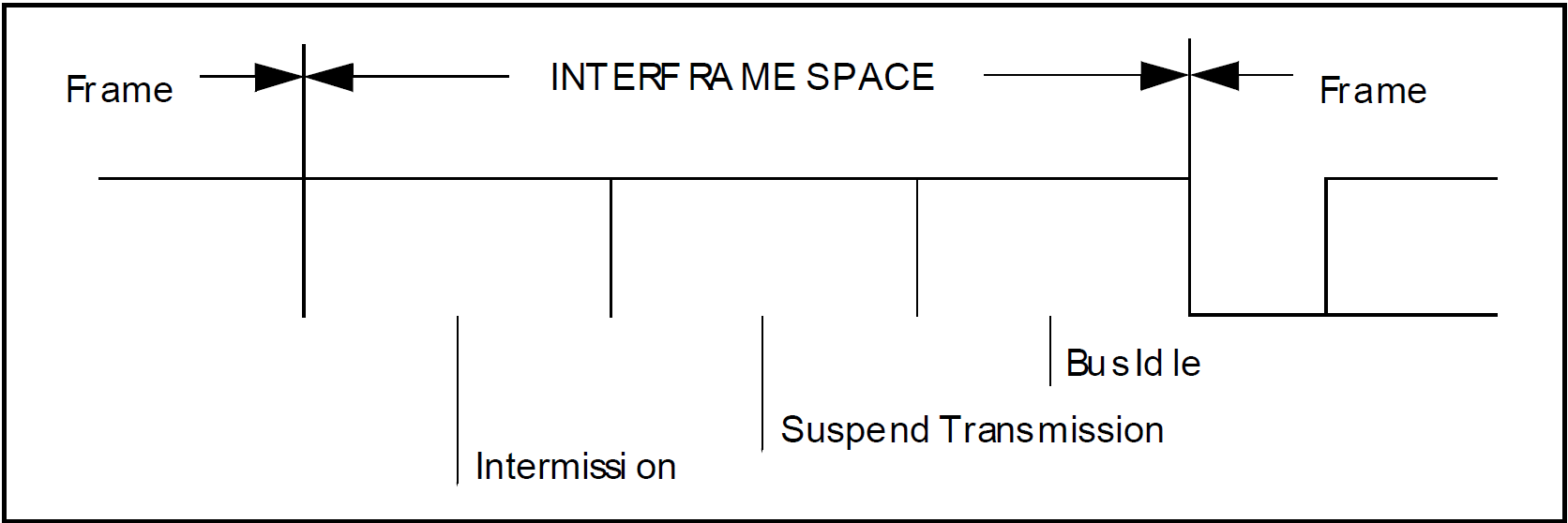
通常为了延时下一个数据帧或远程帧，两个过载帧都会产生。

|  |  |  |
| --- | --- | --- |
| 帧间空间 | INTERFRAME SPACING  数据帧（或远程帧）与其前面帧的隔离是通过帧间空间实现的，无论其前面的帧为何类型（数据帧、远程帧、错误帧、过载帧）。所不同的是，过载帧与错误帧之前没有帧间空间，多个过载帧之间也不是由帧间空间隔离的。  帧间空间包括间歇场、总线空闲的位场。如果“错误被动”的站已作为前一报文的发送器时，则其帧空间除了间歇、总线空闲外，还包括称作挂起传送（SUSPEND TRANSMISSION）的位场。 | |
| 间歇 | 间歇包括3 个“隐性”的位。  间歇期间，所有的站均不允许传送数据帧或远程帧，唯一要做的是标示一个过载条件。 |
| 总线空闲 | 总线空闲的（时间）长度是任意的。只要总线被认定为空闲，任何等待发送信息的站就会访问总线。  在发送其他信息期间，有报文被挂起，对于这样的报文，其传送起始于间歇之后的第一个位。  总线上检测到的“显性”的位可被解释为帧的起始。 |
| 挂起传送 | “错误被动”的站发送报文后，站就在下一报文开始传送之前或总线空闲之前发出8 个“隐性”的位  跟随在间歇的后面。如果与此同时另一站开始发送报文（由另一站引起），则此站就作为这个报文的接收器。 |

对于不是“错误被动” 的站，或者此站已作为前一报文的接收器，其帧间空间如下图所示



对于已作为前一报文发送器的“错误被动”的站，其帧间空间如下图所示：



## CANOpen

|  |  |
| --- | --- |
| 传输次序 | 发送NMT报文时先发送高位，再发送低位； |
| 报文结构 | NMT报文属于CAN**标准数据帧报文**；其数据结构参见《CAN总线协议》  NMT报文的**CAN-ID**、**Data Field、DLC**见下表： |

|  |  |  |  |
| --- | --- | --- | --- |
| NEC报文类型 | 心跳报文 |  | |
| 启动报文 |  | |
| 节点保护 | 节点保护报文 | 从机接收到节点监视报文后，就会发送一条节点保护报文 |
| 节点监视报文 | 由主机发送，指定从机接收 |
| 传输次序 | 发送NEC报文时先发送高位，再发送低位； | | |
| 报文结构 | NEC报文属于CAN**标准数据帧报文**；其数据结构参见《CAN总线协议》  NEC报文的**CAN-ID**、**Data Field、DLC**见下表： | | |

|  |  |
| --- | --- |
| 传输次序 | 发送EMCY报文时先发送高位，再发送低位； |
| 报文结构 | EMCY报文属于CAN**标准数据帧报文**；其数据结构参见《CAN总线协议》  EMCY报文的**CAN-ID**、**Data Field、DLC**见下表： |

|  |  |
| --- | --- |
| 传输次序 | 发送SYNC报文时先发送高位，再发送低位； |
| 报文结构 | SYNC报文属于CAN**标准数据帧报文**；其数据结构参见《CAN总线协议》  SYNC报文的**CAN-ID**、**Data Field、DLC**见下表： |

|  |  |
| --- | --- |
| 传输次序 | 发送TIME报文时先发送高位，再发送低位； |
| 报文结构 | TIME报文属于CAN**标准数据帧报文**；其数据结构参见《CAN总线协议》  TIME报文的**CAN-ID**、**Data Field、DLC**见下表： |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NMT | CAN-ID | 功能码：0000(0x0) | | | |
| Node-ID：本报文发送的目标从机设备的Node-ID；  **Node-ID=0 表明这个报文是广播发送的**  NMT报文具有**最高优先级**的CAN标识符 | | | |
| DLC | **DLC=2** | | | |
| Data | CS  指令  说明符 | 1 | | Service start remote node，用于控制从机进入运行状态 |
| 2 | | Service stop remote node，强制设备进入停止状态 |
| 80 | | Service enter pre-operational，用于控制从机进入预操作状态 |
| 81 | | Service reset node，复位从机设备，  从机设备接收到该命令后重启，进入初始化状态的复位应用状态 |
| 82 | | Service reset communication，复位从机设备，  从机设备接收到该命令后重启，进入初始化状态的复位通信状态 |
|  | 目标设备的节点ID，bit7 = 0 | | |
| 心跳  报文 | CAN-ID | 功能码 | 1110（0x700）  NMT错误对象控制报文对响应时间的要求并不严格，所以分配的优先比较低。 | | |
| Node-ID | 心跳报文生产者设备的Node-ID | | |
| DLC | **DLC=1** |  | | |
| Data | 0x00 | boot up，这是一个启动报文，无论什么设备，一旦从初始化状态进入与操作状态，就会自动发布启动消息报文（boot up），设备将自己的这一状态改变事件（即设备从初始化状态转变到预操作状态这一事件）通知其他设备。 | | |
| 0x04 | Stopped，表示本设备此刻正处于停止状态 | | |
| 0x05 | Operational，表示本设备此刻正处于运行状态 | | |
| 0x7F | Pre-operational，表示本设备此刻正处于预操作状态 | | |
| 节点  保护 | CAN-ID | 功能码 | 1110  NMT错误对象控制报文对响应时间的要求并不严格，所以分配的优先比较低。 | | |
| Node-ID | 发送该报文的**从机**设备的Node-ID | | |
| DLC | **DLC=1** | 节点保护报文 | | |
| **DLC=0** | 节点监视报文，这是一个远程帧报文，RTR=1，数据场=0 | | |
| 数据场 | 高1位**bit7** 是翻转位（Toggle Bit），用于区分当前状态值与历史状态值，从机在两次查询之间改变翻转位，在每次节点保护应答中交替置“0”或者“1”。翻转位在第一次节点保护请求时置为“0”。 | | | |
| Value（hex） | | 指令 | |
| 0x01 或 0x81 | | Disconnected，该状态只有支持扩展boot-up的节点才提供 | |
| 0x02 或 0x82 | | Connecting，该状态只有支持扩展boot-up的节点才提供 | |
| 0x03 或 0x83 | | Preparing，该状态只有支持扩展boot-up的节点才提供 | |
| 0x04 或 0x84 | | Stopped，表示本设备此刻正处于停止状态 | |
| 0x05 或 0x85 | | Operational，表示本设备此刻正处于运行状态 | |
| 0x7F 或 0xFF | | Pre-operational，表示本设备此刻正处于预操作状态 | |
| EMCY | CAN-ID | 功能码 | | 0001（0x80） | |
| Node-ID | |  | |
| DLC | **DLC=8** | | | |
| 数据场 | 0 | | 16位应急错误代码的低8位 | |
| 1 | | 16位应急错误代码的高8位 | |
| 2 | | 8位错误寄存器(Error Register)的值（索引0x1001）  Error Register表示的是设备 **当前的** **一般的** **可供用户快速查看的** 错误状态 | |
| 3~7 | | 制造商特定的错误区域  通常为设备子协议或设备制造商发送的一些附加错误信息。 | |
| SYNC | CAN-ID | 同步对象报文的CAN-ID默认为80h，该值保存在索引为1005h的对象中，且可配置  一般就使用默认的0x80  同步报文不需要设置Node-ID，因为整个网络中只需要一种同步报文 | | | |
| DLC | **DLC=1或0** | | | |
| 数据场 | DLC=0：发送不含同步计数器的同步报文(即不含数据位)，设置溢出值=0  DLC=1：发送包含1字节数据的同步报文（这1字节数据就是同步计数器），设置溢出值=2~240，计数器从1开始递增计数，递增到溢出值后返回到1重新计数；  通过参数Synchronous counter overflow value（索引1019h）来设置溢出值 | | | |
| TIME | CAN-ID | 高4位功能码：0010（0x100） | | | |
| 低7位Node-ID： | | | |
| DLC | **DLC=6** | | | |
| 数据场 |  | | | |

## EMCY16位应急错误代码表

|  |  |  |
| --- | --- | --- |
| xx由相应的设备子协议定义 | | |
| 错误代码 | M/O | 描述 |
| 00xx | M | 错误复位或无错误 |
| 10xx | M | 一般错误 |
| 20xx | O | 电流错误 |
| 21xx | Current, CANopen device input side |
| 22xx | Current inside the CANopen device |
| 23xx | Current, CANopen device output side |
| 30xx | 电压错误 |
| 31xx | Mains voltage |
| 32xx | 设备内部电压 |
| 33xx | Output voltage |
| 40xx | Temperature |
| 41xx | Ambient temperature |
| 42xx | CANopen device temperature |
| 50xx | 硬件错误 |
| 60xx | 软件错误 |
| 61xx | Internal software |
| 62xx | User software |
| 63xx | Data set |
| 70xx | 辅助设备的错误 |
| 80xx | 监视错误 |
| 81xx | 通信错误 |
| 90xx | 外部错误 |
| FFxx | 设备特定的错误 |