

CAN Bootloader 命令及参数说明

1，命令格式

所有数据采用扩展数据帧格式传输，CAN ID 包含节点地址和命令，CAN DATA 包含该命令对应的参数。

保留	节点地址	命令
ID28..ID20	ID19..ID4	ID3..ID0

我们程序中定义的命令如下：

命令	具体值	说明
Erase	0x0	擦除 APP 程序存储区域数据
WriteInfo	0x1	发送连续写数据信息
Write	0x2	发送 APP 数据
Check	0x3	监测当前节点是否在线，并获取固件类型和版本信息
SetBaudRate	0x4	设置节点波特率
Excute	0x5	控制节点执行对应的固件
CmdSuccess	0x8	执行命令成功，节点返回给主控
CmdFaild	0x9	执行命令失败，节点返回给主控

2，命令详解

2.1，Erase

擦除 APP 程序存储区域的数据，该命令带 1 个参数，也就是必须擦除的存储区域大小，单位为字节，目前分为读取 hex 文件或者 bin 文件,一般可以设置为固件文件大小即可，占用 CAN 4 字节数据，格式如下：

待擦除的存储区域大小 FlashSize			
Data[3]	Data[2]	Data[1]	Data[0]
FlashSize[7..0]	FlashSize[15..8]	FlashSize[23..16]	FlashSize[31..24]

主节点发送该命令后，可以通过从节点返回的命令来判断该命令操作是否成功，若成功，则从节点返回 CmdSuccess 命令，若失败则返回 CmdFaild 命令，擦除操作一般比较耗时，在 STM32F4 中尚未测试具体的等待时间,在 TMS230F28335 中测试发现等待时间将近 10s,时间较长,因为是将所有扇区都擦除,此处上位机还需要优化,建议在发送此命令之后尽量多等待一段时间。

2.2，WriteInfo

为了提高固件升级速度，我们采用发送一定量的固件数据之后才进行烧写，比如从节点接收

CAN Bootloader 命令及参数说明 V1.0

到 1024 字节之后才将固件数据写入芯片内部的 Flash 中，所以在传输数据之前，应该先发送该命令，此处依然分为两种不同的情况,分别是 hex 文件和 bin 文件,如果固件文件是 hex 文件,则告知从节点即将缓存的固件数据字节数，以及固件烧写的地址，烧写文件类型，如果固件文件是 bin 文件,则告知从节点即将缓存的固件数据字节数，以及固件烧写的偏移地址，烧写文件类型,因此该命令带三个参数，固件数据地址和从节点应该缓存的固件数据字节数及其烧写文件的类型。

固件数据地址偏移值 AddrOffset			
Data[3]	Data[2]	Data[1]	Data[0]
AddrOffset[7..0]	AddrOffset[15..8]	AddrOffset[23..16]	AddrOffset[31..24]
从节点需要缓存的固件数据字节数 DataNum			
Data[7]	Data[6]	Data[5]	
DataNum[7..0]	DataNum[15..8]	DataNum[23..16]	
烧写文件类型			
Data[4]			
1:hex 文件;2:bin 文件			

该命令同样需要通过从节点的返回命令来判断命令执行是否成功。

2.3, Write

主节点向从节点连续发送固件数据，发送的总数据字节数为 WriteInfo 命令 DataNum 字节大小，由于 CAN 每帧数据最多只能带 8 字节数据，所以可以采用循环发送的方式连续发送大于 8 字节的数据，低地址位数据放入 CAN.Data[0]，高地址位数据依次放入后面的 7 个数据区域，不满 8 字节数据也不用填充，剩余多少就发送多少即可。

当 DataNum 字节数据全部发送完毕之后，从节点应该判断数据的正确性，若数据正确则将数据写入芯片内部的程序存储区域，然后根据最终的状态向主节点返回本次烧写是成功还是失败，然后主节点根据从节点返回的状态值做进一步处理。

若本次操作没有任何问题，则可以进行下一次的操作，依然是先发送 WriteInfo 命令，然后再根据固件数据字节数连续发送 Write 命令，直到所有固件数据全部发送完毕即可。

2.3, Check

检测指定地址的节点工作是否正常，同时获取该节点的固件类型和固件版本信息，主节点发送该命令无需任何参数，从节点接收到该命令后，需要返回获取固件信息的状态命令，若是成功，则需要携带固件类型和固件版本及其当前设备使用的控制器芯片类型三个参数。

固件版本 Version			
次版本号 Minor		主版本号 Major	
Data[3]	Data[2]	Data[1]	Data[0]
Minor[7..0]	Minor[15..8]	Major[7..0]	Major[15..8]
固件类型 Type			
Data[3]	Data[2]	Data[1]	
Type[7..0]	Type[15..8]	Type[23..16]	
固件类型定义:Bootloader 固件类型为： 0x555555; App 固件类型为： 0xAAAAAA			

CAN Bootloader 命令及参数说明 V1.0

控制器芯片类型	
Type[31..24]	
1	TMS320F28335
2	TMS230F2808
3	STM32F407IGT6

2.4, SetBaudRate

更改节点波特率，该命令带一个参数，也就是从节点需要修改的波特率值，从节点更改完波特率值之后，会以新的波特率返回状态值，所以主控节点发送完毕该命令之后也应该尽快切换到新的波特率，若主控节点切换波特率没有从节点切换波特率快，则可以在从节点切换波特率之后加一个延时，否则可能会导致主节点接收不到从节点返回的状态命令。

新波特率值 NewBaudRate			
Data[3]	Data[2]	Data[1]	Data[0]
NewBaudRate[7..0]	NewBaudRate[15..8]	NewBaudRate[23..16]	NewBaudRate[31..24]

2.5, Excute

执行指定类型的固件，该命令带一个参数，也就是固件类型，若该固件类型参数为 Bootloader，而当前节点的固件类型是 App，那么该从节点收到此命令后应立即跳转执行 Bootloader 命令，反之亦然。
该命令没有状态返回，所以可以通过执行 Check 命令来获取固件类型，以此判断固件是否执行成功。

2.6, CmdSuccess

从节点向主节点返回命令执行成功的状态帧，用户可以根据实际情况选择是否携带参数。

2.7, CmdFaild

从节点向主节点返回命令执行失败的状态帧，用户可以根据实际情况选择是否携带参数。