

# STM32 学习笔记

## -----SD 卡 V2.0 协议初始化

《STM32 学习笔记》由 [while \(1\) 世界](#) 会员所总结的学习笔记，供大家参考学习，有问题可以在 while(1) 世界的 STM32 版块提问，共同讨论。

这一笔记是有会员 liklon 总结的学习笔记，如果有问题可以在 [while \(1\) 世界](#) 的 STM32 版块发帖提问！STM32 版块也提供源代码下载。笔记具体类容如下：

现在使用的 4G 的 SD 卡，小于或等于 2G 的卡是属于标准 SD 卡，而大于 2G 的卡小于 32G 的卡是大容量 SD 卡，也就是 SDHC 卡。对于 SDHC 卡的初始化和操作要使用 V2.0 协议。看了几天的 SD 卡 v2.0 协议，现在总结一下啊。首先是一个流程图，这个图在官方资料上有：

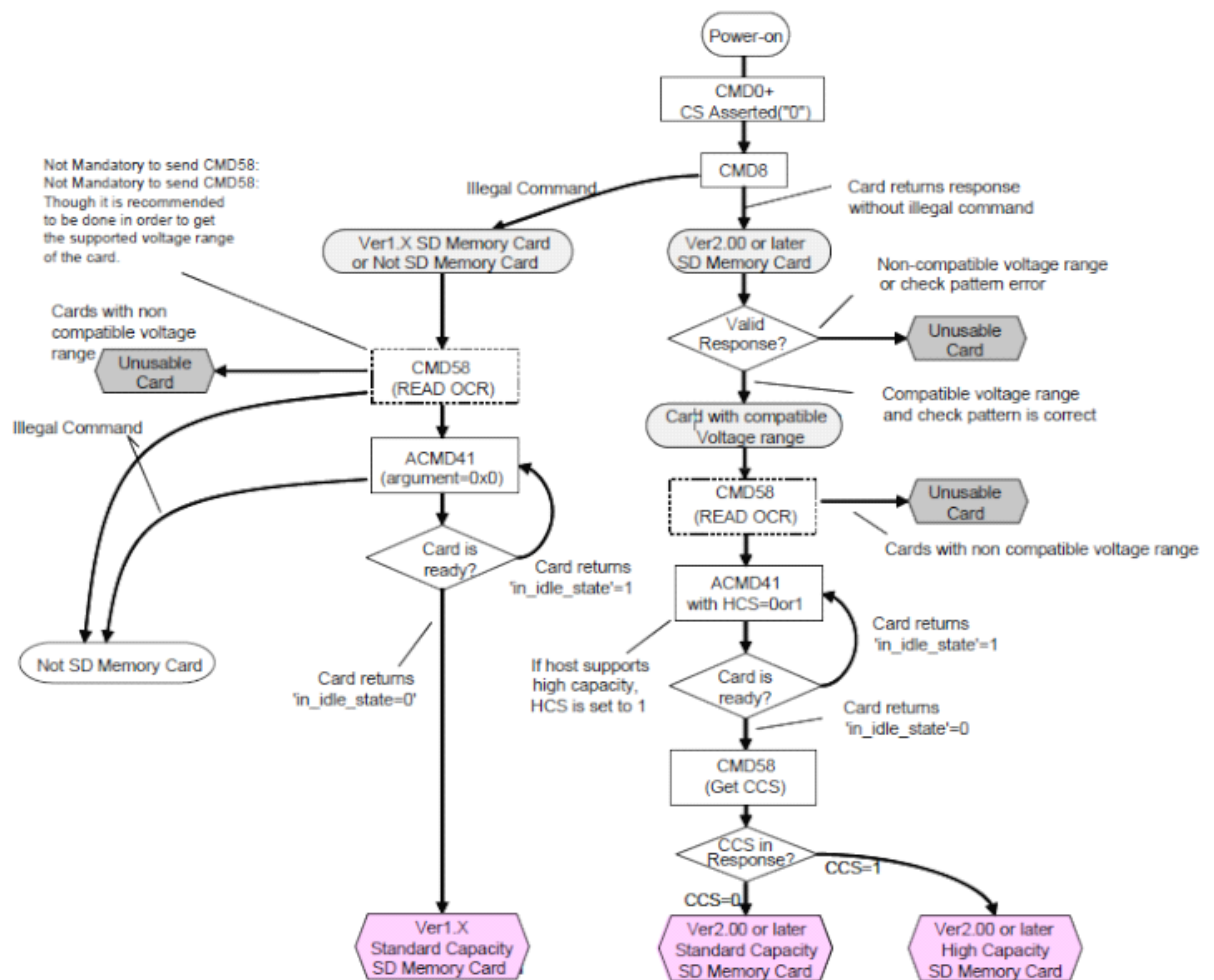


Figure 7-2: SPI Mode Initialization Flow

### SPI 模式下 SD 卡部分操作指令

命令	参数	回应	描述
CM0(0X00)	NONE	R1	复位 SD 卡
CMD9(0X09)	NONE	R1	读取卡特定寄存器
CMD10(0X0A)	NONE	R1	读取卡标志寄存器
CMD16(0X10)	块大小	R1	设置块的大小（字节数）
CMD17(0X11)	地址	R1	读取一块的数据
CMD24(0X18)	地址	R1	写入一块的数据

CMD41(0X29)	NONE	R1	开始卡的初始化
CMD55(0X37)	NONE	R1	引用命令的前命令
CMD59(0X3B)	最后一位有效	R1	设置 CRC 开启（1）或关闭（0）

SD 卡 R1 回应格式：

BIT 7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0	参数错误	地址错误	连续擦除错误	命令 CRC 错误	非法命令	擦除复位	IDLE 状态

卡会根据不同的时候处在不同的状态

Card state	Operation mode
Inactive State	inactive
Idle State	card identification mode
Ready State	
Identification State	
Stand-by State	data transfer mode
Transfer State	
Sending-data State	
Receive-data State	
Programming State	
Disconnect State	

第一步操作： 复位

SD 卡上电后先发送（>74 个时钟），因为 SD 卡有个供电电压上升过程需要大约 64 个时钟，之后的 10 个时钟是用来与 SD 卡同步（参考《例说 STM32》）。参考代码：

```
for(count = 0;count < 15;count++)
```

```
SPI_WriteReadByte(0xff); //产生 74 个以上的脉冲
```

SD 卡默认是 SD 模式，现在用 STM32 去操作，切换为 SPI 模式后更好操作。所以

在片选为低时发送 CMD0，此时卡进入 IDLE 状态，因为 CMD0 回应的命令是 R1，根据上面 R1 的回应格式可以看出我们自需要检查最低位就知道是否处于 IDLE 状态。参考代码：

```
do
{
tmp = SD_WriteCommand(CMD0, 0, 0X95); //发送 SD
count++;
}while((tmp != 0x01) && (count < DISPLAY_COUNT));
```

---

第二步操作发送 CMD8 来分辨卡的类型，是 V2.0 卡还是 V1.0 卡或 MMC 卡，还可以检测 CMD8 响应返回的数据判断是否支持给定的工作电压范围。根据流程图可以看出。

1. 如果 SD 卡支持当前的电压就会返回 R7，并包含 CMD8 的参数部分，其中包括：Check voltage 和 check pattern。
2. 如果 SD 卡不支持当前的工作电压则不会返回任何响应信息，继续处在 IDLE 状态。如果是 V1.0x 的 SD 卡也不会有响应。
3. 在 PLV2.0 (physical layer version 2.0) 下，在首次执行 ACMD41 之前，必须执行 CMD8 指令，用以初始化 SDHC 卡，SDHC 卡根据是否接收到 CMD8 指令来鉴别控制器是否支持 PLV2.0 协议。使用低电压的控制器也必须在 ACMD41 命令之前发送 CMD8，避免可以工作在两种电压模式下的 SD 卡因为没有接收到 CMD8，而默认工作在高电压环境下，被误认为是只支持高电压工作模式。

Bit position	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
Width (bits)	1	1	6	20	4	8	7	1
Value	'0'	'1'	'001000'	'00000h'	x	x	x	'1'
Description	start bit	transmission bit	command index	reserved bits	voltage supplied (VHS)	check pattern	CRC7	end bit

Voltage Supplied	Value Definition
0000b	Not Defined
0001b	2.7-3.6V
0010b	Reserved for Low Voltage Range
0100b	Reserved
1000b	Reserved
Others	Not Defined

Application Note:

It is recommended to use '10101010b' for the 'check pattern'.  
R7 的格式：

7.3.2.6 Format R7

This response token is sent by the card when a SEND\_IF\_COND command (CMD8) is received. The response length is 5 bytes. The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the card operating voltage information and echo back of check pattern in argument and are specified by the same definition as R7 response in SD mode. (Refer to Section 4.9).

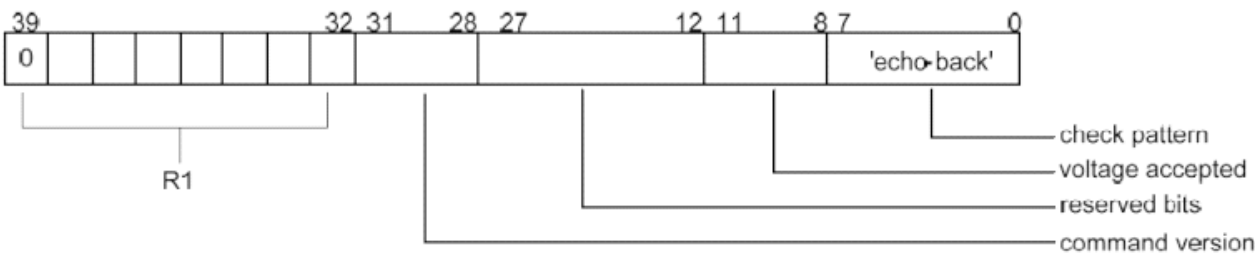


Figure 7-12: R7 Response Format

从上面可以看到，R7 为 5 个字节，在发送 CMD8 后，SD 卡响应，发送回来的第一个字节就是 R1，之后的 4 个字节中就包含了 Check voltage 和 check pattern。下面是仿真的结果：

[0]	0x01
[1]	0x00
[2]	0x00
[3]	0x01
[4]	0xAA

判断 Check voltage 按照如下标准：

voltage accepted	Value Definition
0000b	Not Defined
0001b	2.7-3.6V
0010b	Reserved for Low Voltage Range
0100b	Reserved
1000b	Reserved
Others	Not Defined

Table 4-34: Voltage Accepted in R7

根据流程图就可以看出发送 CMD8 后 SD 卡的类型基本上分了两类。

第三步：由 ACMD41 来初始化 SD 卡，SD 卡的初始化从收到 ACMD41 开始。ACMD 指令的 HCS（Host Capacity Support）位如果设定为 1 的话，表明控制器支持 SDHC 卡，否则表示不支持。

SD 卡初始化和识别过程：

在 CMD8 命令发送之后的 ACMD41 指令其功能有所扩展，在参数里多了 HCS 部分，在响应里面多了 CCS（Card Capacity Status）部分。HCS 参数会被不响应 CMD8 命令的 SD 卡所抛弃。控制器向不响应 CMD8 的卡发送 ACMD41 指令时，HCS 位应

该设置为零 0。如果向 SDHC 卡发送 HCS 位为 0 的 ACMD41 命令，SDHC 卡返回的响应，其 busy 标识位永远为 0，代表忙状态。HCS 标识位用来表明 SD 卡是否已经完成初始化，如果未完成，HCS 为零，否则为 1，如果 HCS 为 0，控制器会重复发送 ACMD41 指令，SD 卡只检查首次接收到的 ACMD41 指令的 HCS 位。

按照流程图，现在发送 ACMD41 可以用来分辨是否为大容量卡。

ACMD INDEX	type	argument	resp	abbreviation	command description
ACMD41	bcr	[31]reserved bit [30]HCS(OCR[30]) [29:24]reserved bits [23:0] VDD Voltage Window(OCR[23:0])	R3	SD_SEND_OP_COND	Sends host capacity support information (HCS) and asks the accessed card to send its operating condition register (OCR) content in the response on the CMD line. HCS is effective when card receives SEND_IF_COND command. Reserved bit shall be set to '0'. CCS bit is assigned to OCR[30].

SD\_WriteCommand(CMD41, 0x40000000, 0X01); //发送 ACMD41，流程图写的如果主机支持大容量就将 HCS 置 1. 所以发送 0x40000000

因为 ACMD41 为应用型指令，所以前面要加上 CMD55 一起使用。

SD\_WriteCommand(CMD55, 0, 0x01);

SD\_WriteCommand(CMD41, 0x40000000, 0X01);

一直发送这两条指令，直到卡准备好。处于 Ready 状态。

所以接收到响应为 0 之后就可以开始读取 OCR 信息。并检测 bit30 位（CCS 位），如果此位是 1 就是大容量卡，如果为 0 则是标准 SD 卡。读取 OCR 的指令是 CMD58，其响应是 R3：

7.3.2.4 Format R3

This response token is sent by the card when a READ\_OCR command is received. The response length is 5 bytes (see Figure 7-11). The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the OCR register.



Figure 7-11: R3 Response Format

前面的 R1 应该为 0。

之后再接收 4 个字节的数据。并判断 bit30 位。就可以判断是 SDV2.0 还是 SDV2.0HC。初始化完成

对于 V1.0 和 MMC 卡的初始化应该照着流程图搞也是一样的。

上面的只是个人的理解，参考了《SD 卡 2.0 协议》，《SD 卡协议学习点滴》，《例说 STM32》。