

Electrical Engineering Stack Exchange is a question and answer site for electronics and electrical engineering professionals, students, and enthusiasts. Join them; it only takes a minute:

Sign up

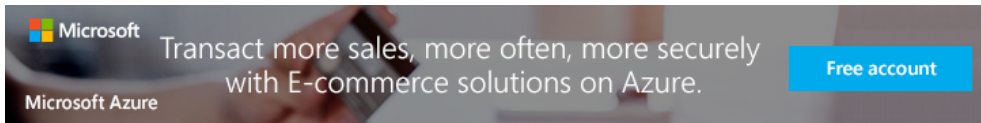
#### Here's how it works:

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top

## What is the correct command sequence for microSD card initialization in SPI?



I am trying to interface a [microSD](#) card (2 GB, Kingston, Sandisk) with a [Silicon Labs C8051F931](#) controller.

I am very much confused about the sequence which I have to follow for initialization. In the book [SD Card Projects Using the PIC Microcontroller](#), page 135 mentions:

The steps to switch the SD card into SPI mode should therefore be as follows:

Power-up.

- Send at least 74 clock pulses to the card with CS and Data Outlines set to logic "1."
- Set CD line low.
- Send 6-byte CMD0 command "40 00 00 00 00 95" to put the card in SPI mode.
- Check R1 response to make sure there are no error bits set.
- Send command CMD1 repeatedly until the "in-idle-state" bit in R1 response is set to "0,"
- and there are no error bits set. The card is now ready for read/write operations.

I tried this, but I am getting 01 even for CDM1. 00 is expected.

Also [here](#) I see a different command sequence where he sends CMD8 after CMD0. But the book says I have to send CMD1.

What is the correct sequence?

microsd

edited Jul 30 '13 at 19:48



Peter Mortensen

1,496 2 13 20

asked Jul 30 '13 at 7:17



gpuguy

546 3 16 37

## 5 Answers

*Actually, most of the info/code you may find on SD initialization is either dated or inaccurate, since it predates SDHC & SDXC by years. The procedure is more complicated nowadays, as it forces you to deal with old hardware in a backwards-compatible way.*

Firstly, as mentioned by others, select a low initial clock rate (generally in the 100 kHz - 400 kHz range; use 400 kHz if possible); you'll be able to switch to higher clock later on, if the device allows for it. While new cards can safely withstand MHz-ish clocking, older will complain (i.e. not communicate or return garbage).

Next thing is that you shouldn't use `CMD1` to initialize SD/SDHC/SDXC cards unless your card doesn't recognize `CMD55` / `ACMD41` ; as said in the SD Card specification:

In any of the cases `CMD1` is not recommended because it may be difficult for the host to distinguish between MultiMediaCard and SD Memory Card.

Some controllers (newer and higher capacity cards mostly) will simply stay in IDLE if you issue `CMD1` to them. You should first issue `CMD8 0x1AA` after the reset ( `CMD0` ), and then try to use `CMD55` + `ACMD41` . *If and only if* that fails, use `CMD1` .

*tl;dr* to initialize the card in SPI mode you should:

1. CMD0 arg: 0x0 , CRC: 0x95 (response: 0x01 ) - note that in case of 0xFF or garbled response you should simply repeat this step; see below for more info.
2. CMD8 arg: 0x000001AA , CRC: 0x87 (response: 0x01 , followed by echo of arg, in this case 0x000001AA ) - while it may seem that this command is optional, it's completely mandatory for newer cards. While 0x1AA is a common arg value here, you may actually pass other values as well; see "Table 7-5: Card Operation for CMD8 in SPI Mode", p. 108 in spec for details.
  - 3a. CMD55 arg: 0x0 , CRC: any, 0x65 actually (response: 0x01 ; CMD55 being the prefix to every ACMD ; if the response is 0x05 , you've got an old card - repeat CMD1 with arg 0x0 [CRC 0xF9 ] instead of CMD55 / ACMD41 )
  - 3b. ACMD41 , arg: 0x40000000 , CRC: any, 0x77 actually (note that this argument assumes the card is a HCS one, which is usually the case; use 0x0 arg [CRC 0xE5 ] for older cards). If response is 0x0 , you're OK; if it's 0x01 , goto 3; if it's 0x05 , see above; if it's neither, *something's wrong with it* (also see below).

Most cards *require* steps 3a/3b (or CMD1 for old cards) to be repeated, usually at least once, *even if you wait some time between them*; i.e. the actual sequence is CMD0 / CMD8 / CMD55 / ACMD41 / CMD55 / ACMD41 (or CMD0 / CMD8 / CMD1 / CMD1 ) - to be sure, try the CMD55 / ACMD41 (or CMD1 if you got 0x05 from them) *n* times (select *n* within your reason; it's actually quite common to have to wait a couple hundred ms if the device is just after the power on, so aim for that), with small delays between the tries if you wish, and assume fail if response 0 doesn't appear (i.e. if the device stays in IDLE mode for some reason). Also, receiving 0xFF from CMD0 is common if a device was in some "strange" state previously (e.g. hung up, got SS deasserted [high], had over-/undervoltage on some pins etc.) - just give it some time, rinse and repeat *n* times. A garbled response to CMD0 is quite OK sometimes - if you sent it a couple of times and the response is still neither 0xFF nor 0x01 , try to go forward with CMD8 . If it works - you're good to go; if it doesn't - it's *probably broken*.

Note that responses that have the MSB set but ain't 0xFF usually suggest that your SPI got a shift in clocking (as a result of e.g. Vcc drop, which happens routinely when you're doing SD hotplugs). To fix it, you can try to completely reset the device (power on/off, deassert/assert SS etc.); it *usually* works.

Also, the spec says

After the last SD Memory Card bus transaction, the host is required, to provide 8 (eight) clock cycles for the card to complete the operation before shutting down the clock.

It could work without it, but since 8 cycles = 1 SPI output byte, it won't hurt much and it's just good to have it.

Note that you should assert SS (aka CS) low *at least* before and after each CMD - it's completely mandatory in case of CMD0 (the device won't turn on without it) and, in reality, required for all other CMD s if you have a standards-compliant SD card. Connecting the card's SS to GND permanently *may seem* to be a good idea if the card is the only SPI client your host will ever connect to, as it would save you both the uC output pin & the need to manage it by code at all, and because the card should assume it's selected all of the time. In reality, some cards (if not most of them) actually expect a high-to-low slope to turn on instead of simply detecting low, and thus get angry if you don't toggle the SS bit at all, and then either lag clocks or spit garbage; some (usually newer) cards should work, some (older) may not, YMMV (yet again). Still, for any more robust SPI configuration (>1 slave device) remember to assert the pin low before any actual transaction with the given SD card.

Besides, while the spec says that only CMD0 and CMD8 should have CRC in SPI mode, *some SD cards (like Transcend ones)* seem to require proper CRC for CMD55 / ACMD41 - if you want to be on the safe side, simply use a precalculated value for them.

Also, while SPI doesn't require pullups/downs by itself, throwing a 47k pullup on MISO may be a good idea; some devices leave their DO pin high-Z under specific circumstances (not initialized e.g.), and floating pins can always be a source of strange problems. **If** your uC has 3.3 Vcc, you may use internal pullups; if it's 5V, don't do that unless your MISO line has a proper 5->3.3V logic translation already.

Further reading:

[How To Use MMC/SDC](#)

[SD Specifications Part 1 Physical Layer Simplified Specification Version 2.00](#)  
[September 25, 2006](#) (section 6.4.1 *Power up*)

vaxquis

797 7 19



Did you find this question interesting? Try our newsletter

Sign up for our newsletter and get our top new questions delivered to your inbox ([see an example](#)).

The specs for SD Cards are available on [sdcard.org](http://sdcard.org). The simplified version left out some details, but you should look for example at the figure 7-2 in part 1, where the initialization sequences are explained for SDHC and SD cards.

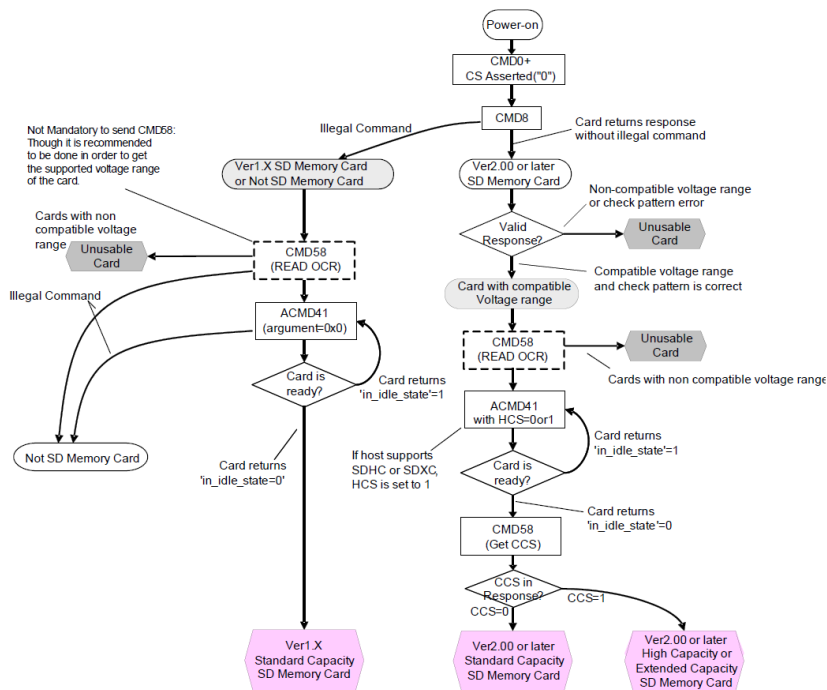
MicroSD cards <= 2 GB can work like older cards, so they should give you `0x00` result for `CMD1` eventually. This may need more than a few retries, as the card can use the external clock from the SPI bus to drive some internal processing.

answered Jul 30 '13 at 11:32

Turbo J

6,364 1 11 20

Adding to @vaxquis excellent answer, I'd like to quote the corresponding chart from the **"Physical Layer Simplified Specification Version 4.10"**, ©Copyright 2001-2013 SD Group (Panasonic, SanDisk, Toshiba) and SD Card Association" (Figure 7-2: SPI Mode Initialization Flow):



Here you can see which commands to send in which order and what the responses tell us about the type of card. I think it is desirable that a device supports as many cards as possible; and as long as it comes to the basic operations of reading and writing blocks of 512 bytes, that should be doable for at least all V1.x and V2.0 SD and HC cards.

answered Feb 3 at 16:55

JimmyB

1,617 7 11

Are you sure your SPI bus is at 400 kHz? The initialization must happen with the SPI bus running at 400 kHz until the SD Card reports it is in the idle state, upon which the SPI bus clock rate may be increased (the exact maximum seems to vary from manufacturer to manufacturer, but it seems like 12 MHz is a safe bet for most cards).

Also, according to this: [http://elm-chan.org/docs/mmc/mmc\\_e.html](http://elm-chan.org/docs/mmc/mmc_e.html) CMD1 is the proper initialization. CMD8 is only required to query the voltage range, which should

not be an issue with non-SDHC ( $\leq 2\text{GB}$ ) cards.

[edited Aug 30 '13 at 0:52](#)

answered Aug 30 '13 at 0:30



[Zuofu](#)

3,498 12 29

---

actually, many SD cards (mostly newer ones, my Sony SR-32C4 32GB being one) won't start at all without CMD8 issued beforehand. Also, the clock is usually not an issue, as long as it's within sensible range. – [vaxquis Jun 4 '16 at 22:39](#)

---

Maybe this is too late, but the response from the card is O.K.! After CMD0, response must be 0x01 - this means that card IS in IDLE state and ready for work. If you have something like 0b00000101 the 1 on the 2-nd place says that this is illegal command and the 1 on the 0-s place says that card IS still in IDLE state and ready for work. If the response is 0x00 it means that card IS NOT in IDLE state and you must send another RESET command.

answered May 19 '14 at 10:46



[Peca](#)

22 2

---

have you read the question at all? OP clearly said I tried this, but I am getting 01 even for CMD1 - getting IDLE in response to CMD1 in **NOT** OK. You're not addressing his real problem with your "answer". – [vaxquis Jun 4 '16 at 14:01](#)

---