Secure Digital (SD) Card Spec and Info

If you're looking for a run down of the features and history of SD or microSD cards, go read the appropriate Wikipedia article (here and here). This page is just going to concern itself with attached these fingernail-sized chunks of storage to commodity microcontrolers.
The SD standard is a mess of extensions and versions and speed classes - in general, just be aware that there are (as of March 2010) three general types of SD cards: SD, SDHC (Secure Digital High Capacity), and SDXC (Secure Digital eXtended Capacity) and that these types are available in the standard SD, the miniSD, and the microSD form factors.
This page will only take about connecting to SD cards that support the SPI interface.

Resources and Credit

Transcend SD card info page
Transcend SD card datasheet sheet (Local Copy)
Transcend microSD card info page
Transcend microSD card datasheet sheet (Local Copy)

SanDisk Secure Digital Card Product Manual, Version 1.9, Document No. 80-13-00169 (Local copy)

~~SD Association SD Card Spec Page~~
~~SD Association SD Specifications, Physical Layer Simplified Specification, Version 2.00 (Local Copy)~~

SD Association SD Simplified Specifications Page
Physical Layer Simplified Specification, Version 3.01 (Local Copy)

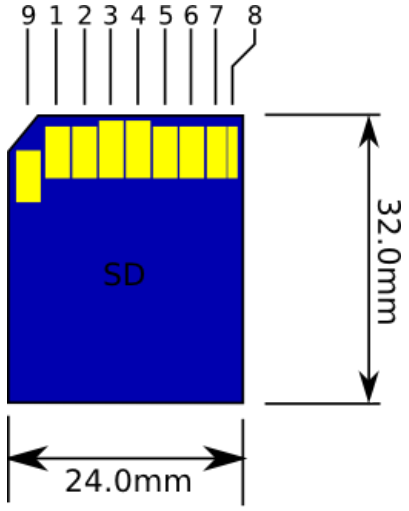
Maxim IC App Note 3969 - "SD Media Format Expands the MAXQ2000's Space for Nonvolatile Data Storage"

Big thanks to **David E.** for a flowchart correction and programing assistance!

pro

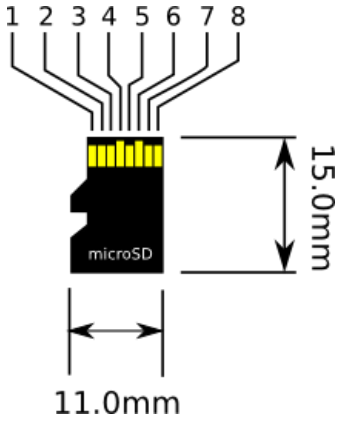
Pinouts

Note: Diagrams somewhat to scale, not exact. Illustrative purposes only. Pin numbering is arbitrary. May not be consistent across all datasheets.
Key: S: Power Supply; I: Input; O: Output; PP: Push-Pull



SD Card

Pin No.	SD Mode			SPI Mode		
	Name	Type	Description	Name	Type	Description
1	CD/DAT	I/O/PP	Card Detect/Data Line [Bit 3]	CS	I	Chip Select (active low)
2	CMD	PP	Command/Response	DI/MOSI	I	Data In/Master Out Slave In
3	Gnd1/Vss1	S	Ground	GND/VSS	S	Ground
4	Vdd	S	Power (2.7V to 3.6V DC)	VDD	S	Power (2.7V to 3.6V DC)
5	CLK	I	Clock	SCLK	I	Clock
6	Gnd2/Vss2	S	Ground	Gnd2/Vss2	S	Ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO/MISO	O/PP	Data Out/Master In Slave Out
8	DAT1	I/O/PP	Data Line [Bit 1]	RSV		Reserved
9	DAT2	I/O/PP	Data Line [Bit 2]	RSV		Reserved



microSD Card

Pin No.	SD Mode			SPI Mode		
	Name	Type	Description	Name	Type	Description
1	DAT2	I/O/PP	Data Line [Bit 2]	RSV		Reserved
2	CD/DAT3	I/O/PP	Card Detect / Data Line [Bit 3]	CS	I	Chip Select
3	CMD	PP	Command/Response	DI/MOSI	I	Data In/Master Out Slave In
4	Vdd	S	Power	Vdd	S	Power
5	CLK	I	Clock	SCLK	I	Clock
6	Gnd/Vss	S	Ground	Gnd/Vss	S	Ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO/MISO	O/PP	Data Out/Master In Slave Out
8	DAT1	I/O/PP	Data Line [Bit 1]	RSV		Reserved

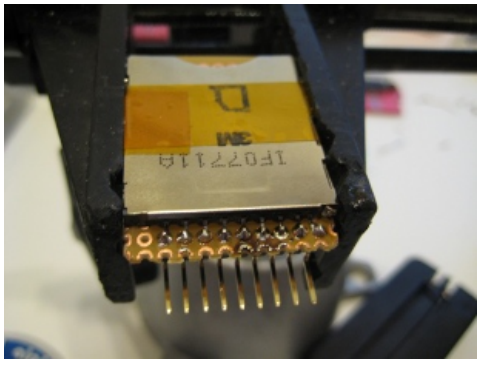
Other Info

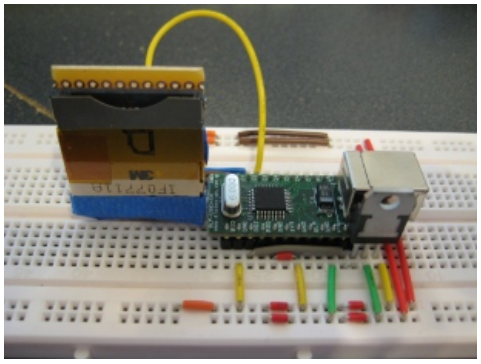
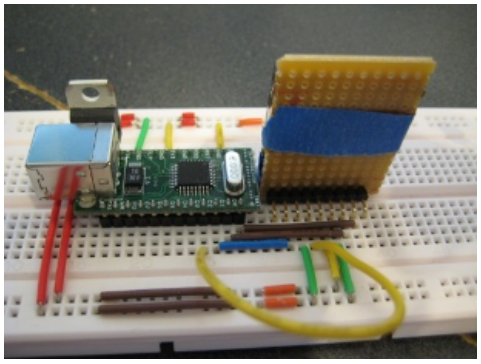
The information and layout of the following table was lifted from the Wikipedia page on the MultiMediaCard card standard:

Type	MMC	SD	microSD

SPI Mode	Optional	Yes	Optional
1 bit mode	Yes	Yes	Yes
4 bit mode	No	Optional	Optional
Xfer clock	0–20 MHz	0–25 MHz - 0–50 MHz	0–25 MHz?
Max Transfer	20 Mbit/s	100 Mbit/s - 200 Mbit/s	100 Mbit/s
Max SPI Transfer	20 Mbit/s	25 Mbit/s	25 Mbit/s
DRM Aval.	No	Yes	Yes

Hardware



I'm trying to prototype an SPI bus interface with a DLP-USB245M (datasheet, local copy) used in bitbang mode with the libftdi library. I'm not going to post details at the moment, because I'm not even sure this approch will work - there is a lot of inconsistancy in the pulse lengths of the outputs. It *shouldn't* matter (being a clocked bus and all) but who knows.
The SD socket is a 3M **SD-RSMT-2-MQ-WF** (3M5646CT-ND on Digikey); datasheet, local copy. Fortuitously, the data pins are 2.50mm (0.098") apart which was close enough to 0.1" to line up with a standard header. I ~~mangled~~ removed the Write Protect and Card Detect pins - they were much closer together and weren't strictly needed.

Serial Peripheral Interface (SPI) Bus

If you don't know what an SPI bus is, go read the Wikipedia artical.
The bus lines are latched on the rising edge of the clock line. The clock line starts low. (CPHA=0 and CPOL=0 according to Freescale's SPI Block Guide and Wikipedia.) The cards internal state machine starts counting clock cycles when the Chip Select line is pulled low. An idle bus (either direction) is denoted with the line being held high. All command tokens and data blocks are built from 8-bit words (aka: bytes).

SD Card SPI Data Transfer Protocol

As mentioned above, all data sent through the SPI bus are built around the byte - some items may have padding, but the host and card will alwasy send/recieve some multiple of 8 bits.
All command tokens are six bytes long. The card will always respond to every command token with a response token of some kind.

Command token format, lifted from the Simplified Physical Layer v3.01 spec:

	Start Bit	Transmission Bit	Command Bit Pattern	Argument	CRC7	End Bit
Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (Bits)	1	1	6	32	7	1
Value	0	1				1

The list of supported commands in SPI mode is a subset of the list of commands the card supports in SD mode. All stuff bits should be set to '0'.
List of commands in SPI mode, lifted from the Simplified Physical Layer v3.01 spec:

CMD INDEX	Bit Pattern	Mnemonic	Argument	Response Format	Command Description
CMD0	000000	GO_IDLE_STATE	[31:0] Stuff Bits	R1	Resets the SD Memory Card
			[31]		Sends host capacity support information

CMD1 (note 1)	000001	SEND_OP_COND	Reserved [30] HCS [29:0] Reserved	R1	and activates the card's initialization process. HCS is effective when card receives SEND_IF_COND command. Reserved bits shall be set to '0'.
CMD6 (note 8)		SWITCH_FUNC	[31] Mode (0 = Check function, 1 = Switch function) [30:24] reserved (All '0') [23:20] reserved for function group 6 (All '0' or 0xF) [19:16] reserved for function group 5 (All '0' or 0xF) [15:12] reserved for function group 4 (All '0' or 0xF) [11:8] reserved for function group 3 (All '0' or 0xF) [7:4] function group 2 for command system [3:0] function group 1 for access mode	R1	Checks switchable function (mode 0) and switches card function (mode 1). See Chapter 4.3.10 of the Physical Layer Simplified Spec.
CMD8 (note 9)	001000	SEND_IF_COND	[31:12] Reserved Bits [11:8] Supply voltage (VHS) [7:0] Check Pattern	R7	Sends SD Memory Card interface condition that includes Host Supply Voltage (VHS) information and asks the accessed card whether card can operate in supplied voltage range. Reserved bits shall be set to '0'.
CMD9		SEND_CSD	[31:0] Stuff Bits	R1	Asks the selected card to send its card-specific data (CSD)
CMD10		SEND_CID	[31:0] Stuff Bits	R1	Asks the selected card to send its card identification (CID)
CMD12		STOP_TRANSMISSION	[31:0] Stuff Bits	R1b (note 5)	Forces the card to stop transmission in Multiple Block Read Operation

CMD13		SEND_STATUS	[31:0] Stuff Bits	R2	Asks the selected card to send its status register.
CMD16	010000	SET_BLOCKLEN	[31:0] Block Length	R1	Sets a block length (in bytes) for all following block commands (read and write) (note 2) of a Standard Capacity Card. Block length of the read and write commands are fixed to 512 bytes in SDHC and SDXC cards. The length of LOCK_UNLOCK command is set by this command in all cards, regardless of capacity.
CMD17		READ_SINGLE_BLOCK	[31:0] Data Address (note 10)	R1	Reads a block of the size selected by SET_BLOCKLEN command (note 3)
CMD18		READ_MULTIPLE_BLOCK	[31:0] Data Address (note 10)	R1	Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command.
CMD24		WRITE_BLOCK	[31:0] Data Address (note 10)	R1	Writes a block of the size selected by the SET_BLOCKLEN command. (note 4)
CMD25		WRITE_MULTIPLE_BLOCK	[31:0] Data Address (note 10)	R1	Continuously writes blocks of data until 'Stop Tran' token is sent (instead of 'Start Block').
CMD27		PROGRAM_CSD	[31:0] Stuff Bits	R1	Programming of the programmable bits of the CSD.
CMD28		SET_WRITE_PROT	[31:0] Data Address	R1b (note 5)	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE). SDHC and SDXC cards do not support this command.
CMD29		CLR_WRITE_PROT	[31:0] Data Address	R1b (note 5)	If the card has write protection features, this command clears the write protection bit of the addressed group. SDHC and SDXC cards do not support this command.
CMD30		SEND_WRITE_PROT	[31:0] Write protect data address	R1	If the card has write protection features, this command asks the card to send the status of the write protection bits. (note 6) SDHC and SDXC cards do not support this command.
CMD32		ERASE_WR_BLK_START_ADDR	[31:0] Data Address (note 10)	R1	Sets the address of the first write block to be erased.
CMD33		ERASE_WR_BLK_END_ADDR	[31:0] Data Address (note 10)	R1	Sets the address of the last write block of the continuous range to be erased.
CMD38		ERASE	[31:0] Stuff Bits	R1b (note 5)	Erases all previously selected write blocks
CMD42		LOCK_UNLOCK	[31:0] Reserved bits (set all to 0)	R1	Used to Set/Reset the Password or lock/unlock the card. A transferred data block includes all the command details - refer to Chapter 4.3.7 of the Physical Layer Simplified Spec v3.01. The size of the Data Block is defined with SET_BLOCK_LEN command. Reserved bits in the argument and in Lock Card Data Structure shall be set to

					0.
CMD55		APP_CMD	[31:0] Stuff Bits	R1	Defines to the card that the next command is an application specific command rather than a standard command.
CMD56		GEN_CMD	[31:1] Stuff Bits [0] RD/WR (note 7)	R1	Used either to transfer a Data Block to the card or to get a Data Block from the card for general purpose/application specific commands. In case of Standard Capacity SD Memory Card, the size of the Data Block shall be defined with SET_BLOCK_LEN command. Block length of this command is fixed to 512-byte in SDHC and SDXC cards.
CMD58		READ_OCR	[31:0] Stuff Bits	R3	Reads the OCR register of a card. CCS bit is assigned to OCR[30].
CMD59		CRC_ON_OFF	[31:1] Stuff Bits [0] CRC option	R1	Turns the CRC option on or off. A '1' turns the option on, a '0' will turn it off.

Notes:
1. CMD1 is valid command for the Thin (1.4mm) Standard Size SD Memory Card only if used after re-initializing a card (not after power on reset).
2. The default block length is as specified in the CSD.
3. The data transferred shall not cross a physical block boundary unless READ_BLK_MISALIGN is set in the CSD.
4. The data transferred shall not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD.
5. R1b: R1 response with an optional trailing busy signal
6. 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line. The last (least significant) bit of the protection bits corresponds to the first addressed group. If the addresses of the last groups are outside the valid range, then the corresponding write protection bits shall be set to zero.
7. RD/WR_: '1' the Host shall get a block of data from the card. '0' the host sends block of data to the card.
8. This command was added in spec version 1.10
9. This command is added in spec version 2.00
10. SDSC Card (CCS=0) uses byte unit address and SDHC and SDXC Cards (CCS=1) use block unit address (512 bytes unit).

Note CMD55 (APP_CMD) - the following table lists most of the application specific commands that may be run after CMD55 in SPI mode. (Commands omitted involved "SD security applications" and don't appear to be freely avalable.)
This list was lifted from the Simplified Physical Layer v3.01 spec.

ACMD INDEX	Bit Pattern	Mnemonic	Argument	Response Format	Command Description
ACMD13		SD_STATUS	[31:0] Stuff Bits	R2	Send the cards status register. Note: The spec says that the card (in SPI mode) will respond with an R2 token, but also states it will return the fields in Table 4-38 (of the Physical Layer Simplified Spec v3.01) - which is a 512 bit block.
ACMD22		SEND_NUM_WR_BLOCKS	[31:0] Stuff Bits	R1	Send the numbers of the well written (without errors) blocks. Responds with 32-bit+CRC data block.
ACMD23		SET_WR_BLK_ERASE_COUNT	[31:23] Stuff Bits [22:0] Number of blocks	R1	Set the number of write blocks to be pre-erased before writing (to be used for faster Multiple Block WR command). '1'=default (one wr block) (note 2).
ACMD41		SD_SEND_OP_COND	[31] Reserved Bit [30] HCS [29:0] Reserved Bits	R1	Sends host capacity support information and activates the card's initialization process. Reserved bits shall be set to '0'.
ACMD42		SET_CLR_CARD_DETECT	[31:1] Stuff Bits [0] set_cd	R1	Connect ('1')/Disconnect ('0') the 50 KOhm pull-up resistor on CS (pin 1) of the card. The pull-up may be used for card detection.
ACMD51		SEND_SCR	[31:0] Stuff Bits	R1	Reads the SD Configuration Register (SCR).

Table notes: (2) Stop Tran Token shall be used to stop the transmission in Write Multiple Block whether the pre-erase (ACMD23) feature is used or not.

When the above commands are issued, the SD card will always respond with one of the following response token.

Response R1

One byte in width.

Bit		
0	In idle state	The card is in idle state and running the initializing process.
1	Erase reset	An erase sequence was cleared before executing because an out of erase sequence command was received.
2	Illegal command	An illegal command code was detected.
3	Communication CRC error	The CRC check of the last command failed.
4	Erase sequence error	An error in the sequence of erase commands occurred.
5	Address error	A misaligned address that did not match the block length was used in the command.
6	Parameter error	The command's argument (e.g. address, block length) was outside the allowed range for this card.
7	MSB	Always Zero

Response R1b
R1b is the same as R1, except the response token may be followed by zero or more bytes set to zero. This is a busy signal - when the user receives a non-zero byte the card is ready for another command.

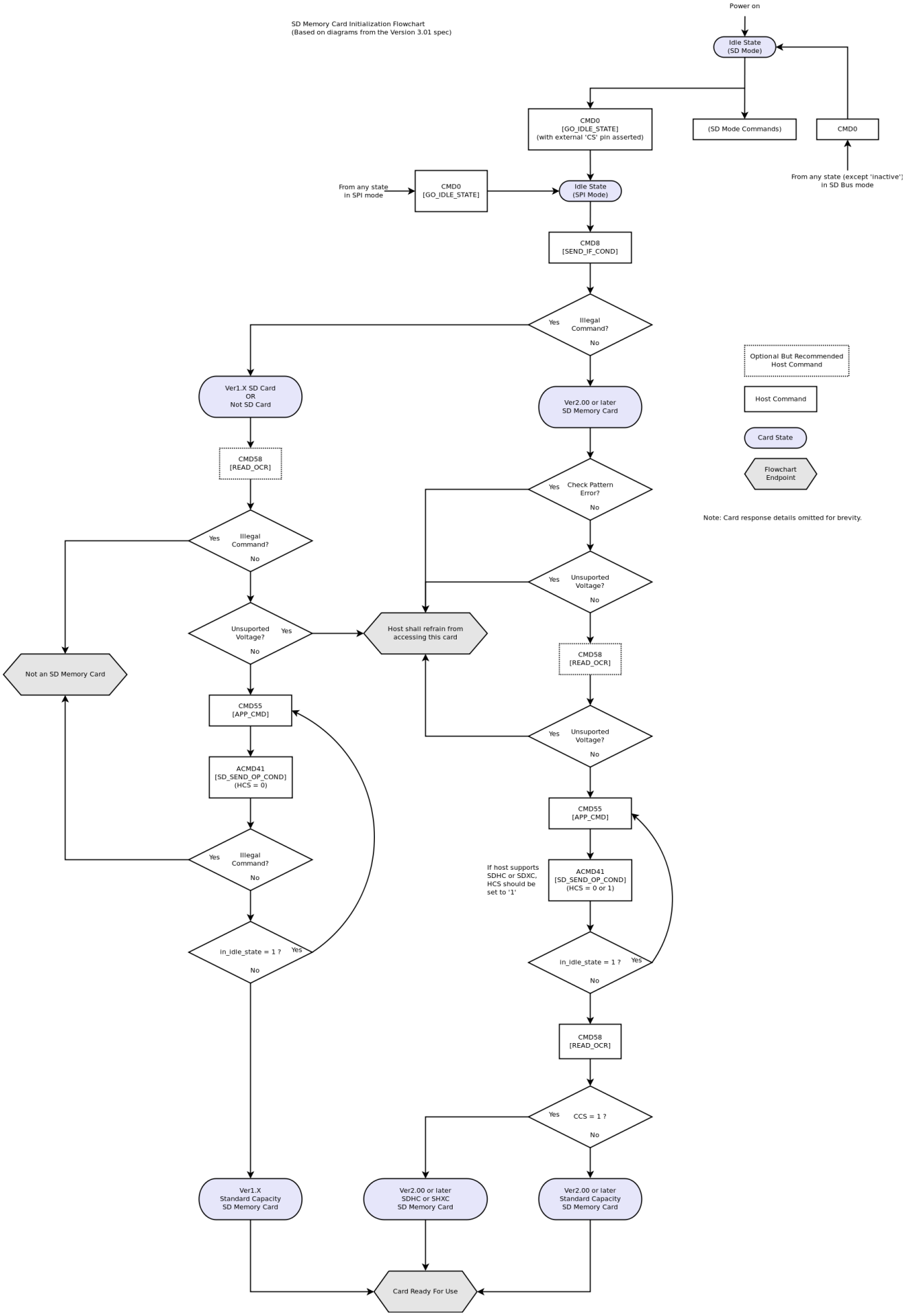
Response R2
Two bytes in width. The first byte sent is identical to R1. The second byte sent is as follows:

Bit		
0	Card is locked	Set when the card is locked by the user. Reset when it is unlocked.
1	Write protect erase skip | lock/unlock command failed	This status bit has two functions overloaded. It is set when the host attempts to erase a write-protected sector or makes a sequence or password errors during card lock/unlock operation.
2	Error	A general or an unknown error occurred during the operation.
3	CC error	Internal card controller error.
4	Card ECC failed	Card internal ECC was applied but failed to correct the data.
5	Write protect violation	The command tried to write a write-protected block.
6	Erase param	An invalid selection for erase, sectors or groups.
7	out of range | csd overwrite	

Response R3
Five bytes in width. The first byte sent is identical to R1. The following four bytes are the contents of the OCR register.

Response R7
Five bytes in width. This is the response token is sent by the card when a SEND_IF_COND command (CMD8) is received.
[TODO: Figure this out, ugh.]

Who wants a flowchart?! Everybody loooves flowcharts!



And that is how you bring an SD card from power on to initialized and ready for use. (Click image to embiggen.)
I'm not going to blow smoke up your tailpipe and tell you it's better than it looks, because it's not. The good news is if you know exactly what you're plugging in you can skip the irrelevant bits - you don't need to implement the *whole* tree for a simple personal project. Some cards may let you get away with far fewer setup commands than depicted, in fact some may let you start slinging data immediately after CMD0 with CS asserted - your mileage may vary though.

Still to do:

- Flow charts and description of tokens for read/write commands.
- Actually test this stuff.
- Detail registers on card
