Secure Digital

From Wikipedia, the free encyclopedia

Secure Digital (**SD**) is a non-volatile memory card format for use in portable devices, such as mobile phones, digital cameras, GPS navigation devices, and tablet computers.

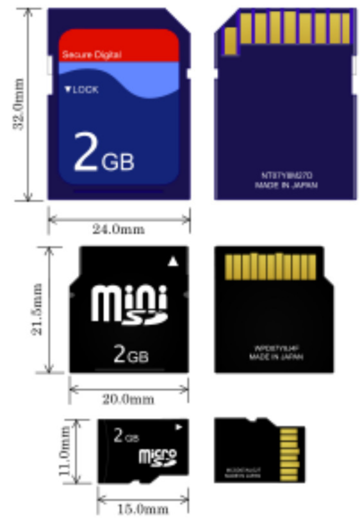
The Secure Digital standard was introduced in 1999 as an evolutionary improvement over MultiMediaCards (MMC). The Secure Digital standard is maintained by the SD Card Association (SDA). SD technologies have been implemented in more than 400 brands across dozens of product categories and more than 8,000 models.[1]

The Secure Digital format includes four card families available in three different form factors. The four families are the original Standard-Capacity (SDSC), the High-Capacity (SDHC), the eXtended-Capacity (SDXC), and the SDIO, which combines input/output functions with data storage.[2][3][4] The three form factors are the original size, the "mini" size, and the "micro" size. Electrically passive adapters allow the use of a smaller card in a host device built to hold a larger card.

There are many combinations of form factors and device families, although as of 2014, the prevailing formats are full- or micro-size SDHC and full/micro SDXC.

The SDA uses several trademarked logos to enforce compliance with its specifications and assure users of compatibility.[5]

SD, SDHC, SDXC



SD (top), miniSD, microSD cards

Media type	Memory card
Capacity	SDSC (SD): 1 MB to 2 GB, some 4 GB available. SDHC: 4 GB to 32 GB SDXC: ≥32 GB to 2 TB
Developed by	SD Card Association
Dimensions	Standard: 32.0×24.0×2.1 mm (1.26×0.94×0.083 in) Mini: 21.5×20.0×1.4 mm (0.85×0.79×0.055 in) Micro: 15.0×11.0×1.0 mm (0.59×0.43×0.039 in)
Weight	Standard: ~2 g Mini: ~0.8 g Micro: ~0.25 g
Usage	Portable devices, including digital cameras and handheld computers
Extended from	MultiMediaCard (MMC)

Contents

Overview

SD

The first-generation Secure Digital (SDSC or Secure Digital Standard Capacity) card was developed to improve on the Multimedia Card (MMC) standard. The SD cards changed the MMC design in several ways:

- Asymmetrical slots in the sides of the SD card prevent inserting it upside down, while an MMC goes in most of the way but makes no contact if inverted.
- Most SD cards are 2.1 mm (0.083 inches) thick, compared to 1.4 mm (0.055 inches) for MMCs. The SD specification defines a card called **Thin SD** with a thickness of 1.4 mm, but they are rare, as the SDA went on to define even smaller form factors.
- The card's electrical contacts are recessed beneath the surface of the card, protecting them from contact with a user's fingers.
- The SD specification envisioned capacities and transfer rates exceeding those of MMC, and these have both grown over time. For a comparison table, see below.

Full-sized SD cards do not fit in the slimmer MMC slots, and there are other issues that affect the ability to use one format in a host device designed for the other.

SDHC

The Secure Digital High Capacity (SDHC) format, announced in January 2006 and defined in Version 2.0 of the SD specification, supports cards with capacities up to 32 GB.[2] The SDHC trademark is licensed to ensure compatibility.[6]

Official SDHC logo

SDHC cards are physically and electrically identical to standard-capacity SD cards (SDSC). The major compatibility issues between SDHC and SDSC cards are the redefinition of the Card-Specific Data (CSD) register in Version 2.0 (see below), and the fact that SDHC cards are shipped preformatted with the FAT32 file system.

Version 2.0 also introduces a High-speed bus mode for both SDSC and SDHC cards, which doubles the original Standard Speed clock to produce 25 Mbyte/s.

SDHC host devices are required to accept older SD cards.[1] However, older host devices do not recognize SDHC or SDXC memory cards, although some devices can do so through a firmware upgrade.[7] Older Windows operating systems released before Windows 7 require patches or service packs to support access to SDHC cards.[8][9][10]

SDXC

The Secure Digital eXtended Capacity (SDXC) format, announced in January 2009 and defined in Version 3.01 of the SD specification, supports cards up to 2 TB (2048 GB), compared to a limit of 32 GB for SDHC cards in the SD 2.0 specification. SDXC adopts Microsoft's exFAT file system as mandatory feature.

Official SDXC logo

The 3.0 specification also introduced the Ultra-High Speed (UHS) bus, with interface speeds from 50 Mbyte/s to 104 Mbyte/s for 4-bit UHS-I bus.

Version 4.0, introduced in June 2011, allows speeds of 156 Mbyte/s to 312 Mbyte/s over 8-bit UHS-II bus, which requires additional row of physical pins.

Due to proprietary nature of exFAT file system, BSD and Linux systems do not support it out of the box. There is an open source implementation available [11] as a FUSE module. The user may reformat the card to contain a different file system (see below).

Windows XP SP2 and later and Apple Mac OS X 10.6.5 and later support exFAT and SDXC.[12][13][14]

exFAT filesystem

SDXC cards are pre-formatted with Microsoft's proprietary and patented exFAT file system. Microsoft does not publish the specifications of exFAT and its use requires a non-free license, so many alternative or older operating systems do not support exFAT for technical or legal reasons, even if they can support the underlying hardware such as SDXC card readers. This makes SDXC cards unsuitable as a universal exchange medium, as an SDXC card that uses exFAT would not be usable in all SD card host devices.

For greater portability, SDXC cards can be reformatted to use alternate file systems, for example FAT32 which supports volumes up to the SDXC's maximum theoretical capacity of 2 TB, and can be used in a SDHC host device that can handle volumes larger than 32 GB. However using alternative file systems bears a risk of accidental loss of data, as the host device may treat a card with unrecognized file system as blank or damaged and reformat the card.

SD Card Association provides a formatting utility for Windows and Mac OS X which can be used to check and format SD, SDHC, and SDXC cards. [15]

Ultra-High Speed bus

The Ultra-High Speed (UHS) bus is available on some SDHC and SDXC cards.[16][17] The following ultra-high speeds are specified:

- **UHS-I** cards, specified in SD Version 3.01,[18] support a clock frequency of 100 MHz (a quadrupling of the original "Default Speed"), which in four-bit transfer mode could transfer 50 MB/s. UHS-I cards declared as **UHS104** (SDR104) also support a clock frequency of 208 MHz, which could transfer 104 MB/s. UHS-I is the fastest class for which products are currently available.[19]
- Double data rate operation at 50 MHz (DDR50) is also specified in Version 3.01, and is mandatory for microSDHC and microSDXC cards labeled as UHS-I. In this mode, four bits are transferred when the clock signal rises and another four bits when it falls, transferring an entire byte on each full clock cycle.
- **UHS-II** cards, defined in Version 4.0, further raise the data transfer rate to a theoretical maximum of 312 MB/s using additional row of pins.[20][21]

UHS memory cards work best with UHS host devices. The combination lets the user record HD resolution videos with tapeless camcorders while performing other functions. It is also suitable for real-time broadcasts and capturing large HD videos.

Cards that comply with UHS show Roman numerals 'I' or 'II' next to the SD card logo,[22] and report this capability to the host device. Use of UHS requires that the host device command the card to drop from 3.3-volt to 1.8-volt operation and select the 4-bit transfer mode.

SDIO

A SDIO (Secure Digital Input Output) card is an extension of the SD specification to cover I/O functions. SDIO cards are only fully functional in host devices designed to support their input-output functions (typically PDAs like the Palm Treo, but occasionally laptops or mobile phones). These devices can use the SD slot to support GPS receivers, modems, barcode readers, FM radio tuners, TV tuners, RFID readers, digital cameras, and interfaces to Wi-Fi, Bluetooth, Ethernet, and IrDA. Many other SDIO devices have been proposed, but it is now more common for I/O devices to connect using the USB interface.

Camera using the SDIO interface to connect to some HP iPAQ devices

SDIO cards support most of the memory commands of SD cards. SDIO cards can be structured as 8 logical cards, although currently, the typical way that an SDIO card uses this capability is to structure itself as one I/O card and one memory card.

The SDIO and SD interfaces are mechanically and electrically identical. Host devices built for SDIO cards generally accept SD memory cards without I/O functions. However, the reverse is not true, because host devices need suitable drivers and applications to support the card's I/O functions. For example, an HP SDIO camera usually does not work with PDAs that do not list it as an accessory. Inserting an SDIO card into any SD slot causes no physical damage nor disruption to the host device, but users may be frustrated that the SDIO card does not function fully when inserted into a seemingly compatible slot. (Bluetooth devices exhibit comparable compatibility issues, although to a lesser extent thanks to standardized Bluetooth profiles.)

Compatibility

Host devices that comply with newer versions of the specification provide backward compatibility and accept older SD cards.[23] For example, SDXC host devices accept all previous families of SD memory cards, and SDHC host devices also accept standard SD cards.

Older host devices generally do not support newer card formats, and even when they might support the bus interface used by the card, [3] there are several factors that arise:

- A newer card may offer greater capacity than the host device can handle (over 4 GB for SDHC, over 32 GB for SDXC).
- A newer card may use a file system the host device cannot navigate (FAT32 for SDHC, exFAT for SDXC)
- Use of an SDIO card requires the host device be designed for the input/output functions the card provides.
- The hardware interface of the card was changed starting with the Version 2.0 (new bus clocks) and SDHC family (UHS bus)
- Some vendors produced SDSC cards above 1 GB before the SDA had standardized a method of doing so.

Speeds

An SD card's speed is measured by how quickly information can be read from, or written to, the card. In applications that require sustained write throughput, such as video recording, the device might not perform satisfactorily if the SD card's class rating falls below a particular speed. For example, a high-definition camcorder may require a card of not less than Class 6, suffering dropouts or corrupted video if a slower card is used. Digital cameras with slow cards may take a noticeable time after taking a photograph before being ready for the next, while the camera writes the first picture.

With early SD cards the speed was specified as a "times" ("×") rating, which compared the average speed of reading data to that of the original CD-ROM drive. This was superseded by the **Speed Class Rating**, which guarantees a minimum rate at which data can be written to the card.

The newer families of SD card improve card speed by increasing the bus rate (the frequency of the clock signal that strobes information into and out of the card). Whatever the bus rate, the card can signal to the host that it is "busy" until a read or a write operation is complete. Compliance with a higher speed rating is a guarantee that the card limits its use of the "busy" indication.

Bus Interface	Card logo	Bus logo	Bus Speed	Spec Version
Normal Speed		—	12,5 Mbyte/s	1.01
High Speed			25 Mbyte/s	2.00
UHS-I		I	50 Mbyte/s (SDR50, DDR50) 104 Mbyte/s (SDR104)	3.01
UHS-II		II	156 Mbyte/s (FD156) 312 Mbyte/s (HD312)	4.00/4.10[24]

Speed class rating

The SD Association defines standard speed classes indicating minimum performance to record video. Both read and write speeds must exceed the specified value. The specification defines these classes in terms of performance curves that translate into the following minimum read-write performance levels on an empty card and suitability for different applications:[18][25]

Sandisk microSDXC card with UHS-I and UHS Speed Class 1 markings and no Speed Class marking

Class	Minimum performance	Application
Class 2	2 MB/s	SD video recording
Class 4	4 MB/s	High-definition video (HD) recording including Full HD (from 720p to 1080p/1080i)
Class 6	6 MB/s	
Class 10	10 MB/s	Full HD (1080p) video recording and consecutive recording of HD stills (high-speed data bus)
UHS Class 1 (U1)	10 MB/s	Real-time broadcasts and large HD video files (UHS bus)
UHS Class 3 (U3)	30 MB/s	4K video files (UHS bus)

Speed classes 2, 4, and 6 assert that the card supports the respective number of megabytes per second as a minimum sustained write speed for a card in a fragmented state. Class 10 asserts that the card supports 10 MB/s as a minimum non-fragmented sequential write speed and uses a High-speed bus mode.[18] The host device can read a card's speed class and warn the user if the card reports a speed class that falls below an application's minimum need.[18] By comparison, the older "×" rating measured maximum speed under ideal conditions, and was vague as to whether this was read speed or write speed.

UHS-I and UHS-II cards can use *UHS Speed Class* rating with two possible grades: class 1 for minimum read/write performance of at least 10 MB/s ('U1' symbol featuring number 1 inside 'U') and class 3 for minimum write performance of 30 MB/s ('U3' symbol featuring 3 inside 'U'), targeted at recording 4K video. [26] Before November 2013, the rating was branded *UHS Speed Grade* and contained grades 0 (no symbol) and 1 ('U1' symbol). Manufacturers can also display standard speed class symbol (a digit encircled with 'C') alongside UHS speed class.

The speed class rating does not totally characterize card performance. Different cards of the same class may vary considerably while meeting class specifications. A card's speed depends on many factors, including:

- The frequency of soft errors that the card's controller must re-try
- The fact that, on most cards, writing data requires the controller to read and erase a larger region, then rewrite that entire region with the desired part changed
- File fragmentation: where there is not sufficient space for a file to be recorded in a contiguous region, it is split into non-contiguous fragments. This does not cause rotational or head-movement delays as with electromechanical hard drives, but may decrease speed; for instance, by requiring additional reads and computation to determine where on the card the file's next fragment is stored.

In addition, speed may vary markedly between writing a large amount of data to a single file (sequential access, as when a digital camera records large photographs or videos) and writing a large number of small files (a random-access use common in smartphones). One study found that, in this random-access use, some Class 2 cards achieved a write speed of 1.38 MB/s, while all cards tested of Class 6 or greater (and some of lower Classes; lower Class does not *necessarily* mean better small-file performance), including those from major manufacturers, were over 100 times slower.[27]

“×” rating

Main article: X (writing speed)

The “×” rating, made obsolete by speed classes, is a multiple of the standard CD-ROM drive speed of 150 KB/s (approximately 1.23 Mbit/s). Basic cards transfer data at up to six times (6×) the CD-ROM speed; that is, 900 KB/s or 7.37 Mbit/s. The 2.0 specification defines speeds up to 200×, but is not as specific as Speed Classes are on how to measure speed. Manufacturers may report best-case speeds and may report the card's fastest read speed, which is typically faster than the write speed. Some vendors, including Transcend and Kingston report their cards' write speed.[28] When a card lists both a speed class and an “×” rating, the latter may be assumed a read speed only.

Rating	Approx. (MB/s)	Comparable Speed Class
16×	2.34	2 (13×)
32×	4.69	4 (27×)
48×	7.03	6 (40×)
100×	14.6	10 (67×)

Features

Card security

Cards can protect their contents from erasure or modification, prevent access by non-authorised users, and protect copyrighted content using digital rights management (DRM).

Commands to disable writes

The host device can command the SD card to become read-only (to reject subsequent commands to write information to it). There are both reversible and irreversible host commands that achieve this.

Write-protect notch

The user can designate most full-size SD cards as read-only by use of a sliding tab that covers a notch in the card. (The miniSD and microSD formats do not support a write protection notch.)

When looking at the SD card from the top, the right side (the side with the beveled corner) must be notched.

On the left side there may be a write-protection notch. If the notch is omitted, the card can be read and written. If the card is notched, it is read-only. If the card has a notch and a sliding tab which covers the notch, the user can slide the tab upward (toward the contacts) to declare the card read/write, or downward to declare it read-only.

The presence of a notch, and the presence and position of a tab, have no effect on the SD card's operation. A host device that supports write protection should refuse to write to an SD card that is designated read-only in this way. Some host devices do not support write protection, which is an optional feature of the SD specification. Drivers and devices that do obey a read-only indication may give the user a way to override it.

Cards sold with content which must not be altered are permanently marked read-only by having a notch and no sliding tab.

Card password

A host device can lock an SD card using a password of up to 16 bytes, typically supplied by the user. A locked card interacts normally with the host device except that it rejects commands to read and write data. A locked card can be unlocked only by providing the same password. The host device can, after supplying the old password, specify a new password or disable locking. Without the password (typically, in the case that the user forgets the password), the host device can command the card to erase all the data on the card for future re-use (except card data under DRM), but there is no way to gain access to the existing data.

MicroSD to SD adapter (left), microSD to miniSD adapter (middle), microSD card (right)

DRM copy-protection

All cards incorporate DRM copy-protection. Roughly 10% of the storage capacity of an SD card is a "Protected Area" not available to the user, but is used by the on-card processor to verify the identity of an application program that it then allows to read protected content. The card prohibits other accesses, such as users trying to make copies of protected files. (Reformatting an SD card may erase this Protected Area; see Risks of reformatting below.)

The DRM scheme embedded in the SD cards is the Content Protection for Recordable Media (CPRM or CPPM) specification of the 4C Entity, which features the Cryptomeria cipher (also termed *C2*). The specification is kept secret and is accessible only to licensees. The scheme has not been broken or cracked; however, this feature of SD cards is rarely used to protect content. DVD-Audio uses the same DRM scheme.

Windows Media files can be DRM-encoded so as to make use of the SD card's DRM abilities.[*citation needed*]

Windows Phone 7 devices use SD cards designed for access only by the phone manufacturer or mobile provider. An SD card inserted into the phone underneath the battery compartment becomes locked "to the phone with an automatically generated key" so that "the SD card cannot be read by another phone, device, or PC".[29] Symbian devices, however, are some of the few that can perform the necessary low-level format operations on locked SD cards. It is therefore possible to use a device such as the Nokia N8 to reformat the card for subsequent use in other devices.[30]

A *Super Digital* product formerly[31] made by Super Talent was essentially the SD card without the DRM features.

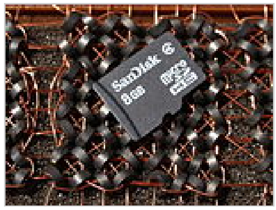
Vendor enhancements

Vendors have sought to differentiate their products in the market through various vendor-specific features:

SD cards with dual-interface SD and USB connections

- **Integrated Wi-Fi** — Several companies produce SD cards with built-in Wi-Fi transceivers supporting static security (WEP 40; 104; and 128, WPA-PSK, and WPA2-PSK). The card lets any digital camera with an SD slot transmit captured images over a wireless network, or store the images on the card's memory until it is in range of a wireless network. Examples include: Eye-Fi / SanDisk, Transcend, Toshiba, PQI Air Card and ez Share.[32] Some models geotag their pictures.
- **Pre-loaded content** — In 2006, SanDisk announced Gruvi, a microSD card with extra digital rights management (DRM) features, which they intended as a medium for publishing content. SanDisk again announced pre-loaded cards in 2008, under the slotMusic name, this time not using any of the DRM capabilities of the SD card.[33] In 2011, SanDisk offered various collections of 1000 songs on a single slotMusic card for about $40,[34] now restricted to compatible devices and without the ability to copy the files.
- **Integrated USB connector** — The SanDisk **SD Plus** product can be plugged directly into a USB port without needing a USB card reader.[35] Other companies introduced comparable products, such as the **Duo SD** product of OCZ Technology and the **3 Way** (microSDHC, SDHC, and USB) product of A-DATA, which was available in 2008 only.
- **Different colors** — SanDisk has used various colors of plastic or adhesive label, including a "gaming" line in translucent plastic colors that indicated the card's capacity.
- **Integrated display** — In 2006, A-DATA announced a **Super Info SD** card with a digital display that provided a two-character label and showed the amount of unused memory on the card.[36]

History



This microSDHC card holds 8 billion bytes. It rests on a section of a magnetic-core memory (used until the 1970s) that holds eight bytes using 64 cores. (This SD card's capacity is one billion times larger)

In 1999, SanDisk, Matsushita, and Toshiba agreed to develop and market the Secure Digital (SD) Memory Card, which was a development of the MultiMediaCard (MMC). The new card provided both digital rights management (DRM) up to the Secure Digital Music Initiative (SDMI) standard, and a high memory density for the time.

The new format was designed to compete with the Memory Stick, a DRM product that Sony released the prior year. It was mistakenly predicted that DRM features[37] would be widely used due to pressure from music and other media suppliers to prevent piracy.

The trademarked *SD* logo was originally developed for the Super Density Disc, which was the unsuccessful Toshiba entry in the DVD format war. This is why the *D* resembles an optical disc.

At the 2000 Consumer Electronics Show (CES) trade show, the three companies announced the creation of the SD Card Association (SDA) to promote SD cards. The SD Card Association's headquarters are in San Ramon, California, United States and it comprises some 30 product manufacturers that make interoperable memory cards and devices. Early samples of the SD Card were available in the first quarter of 2000, with production quantities of 32 and 64 MB cards available 3 months later.

Mini- and micro-cards

In March 2003, SanDisk Corporation announced the introduction of the **miniSD** and demonstrated it at CeBIT 2003.[38] The SDA adopted the miniSD card in 2003 as a small form factor extension to the SD card standard. While the new cards were designed especially for mobile phones, they are usually packaged with a miniSD adapter that provides compatibility with a standard SD memory card slot.

At CTIA Wireless 2005, the SDA announced the small **microSD** form factor (and SDHC, with capacities in excess of 2 GB and a minimum sustained read and write speed of 17.6 Mbit/s). SanDisk had conceived microSD when its CTO and the CTO of Motorola concluded that current memory cards were too large for mobile phones. The card was originally called T-Flash, but just before product launch, T-Mobile sent a cease and desist order to SanDisk claiming that T-Mobile owned the trademark on T-(anything) [*citation needed*], and the name was changed to TransFlash. TransFlash and microSD cards are the same; each can be used in devices made for the other.[39] SanDisk induced the SDA to administer the microSD standard. The SDA approved the final microSD specification on July 13, 2005. Initially, microSD cards were available in capacities of 32, 64, and 128 MB.

In September, 2006, SanDisk announced the 4 GB miniSDHC.[40] Like the SD and SDHC, the miniSDHC card has the same form factor as the older miniSD card but the HC card requires HC support built into the host device. Devices that support miniSDHC work with miniSD and miniSDHC, but devices without specific support for miniSDHC work only with the older miniSD card.

SDIO, SDHC and SDXC

In April 2006, the SDA released a detailed specification for the non-security related parts of the SD memory card standard and for the Secure Digital Input Output (SDIO) cards and the standard SD host controller.

SDHC format has been announced in January 2006 and brought improvements such as 32 GB storage capacity and mandatory support for FAT32 filesystem.

In January 2009, the SDA announced the SDXC family, which supports cards up to 2 TB and speeds up to 300 MB/s. It features mandatory support for exFAT filesystem.

SDXC was announced at Consumer Electronics Show (CES) 2009 (January 7–10, 2009). At the same show, SanDisk and Sony also announced a comparable Memory Stick XC variant with the same 2 TB maximum as SDXC,[41] and Panasonic announced plans to produce 64 GB SDXC cards.[42]

On March 6, 2009, Pretec introduced the first SDXC card,[43] a 32 GB card with a read/write speed of 400 Mbit/s. But it was not until early 2010 that compatible host devices came onto the market, including Sony's Handycam HDR-CX55V camcorder, Canon's EOS 550D (also known as Rebel T2i) Digital SLR camera,[44] a USB card reader from Panasonic, and an integrated SDXC card reader from JMicron.[45] The earliest laptops to integrate SDXC card readers relied on a USB 2.0 bus, which does not have the bandwidth to support SDXC at full speed.[46]

Also in early 2010, commercial SDXC cards appeared from Toshiba (64 GB),[47][48] Panasonic (64 GB and 48 GB),[49] and SanDisk (64 GB).[50] In early 2011, Centon Electronics, Inc. (64 GB and 128 GB) and Lexar (128 GB) began shipping SDXC cards rated at Speed Class 10.[51] Pretec offered cards from 8 GB to 128 GB rated at Speed Class 16.[52]

In September 2011, SanDisk released a 64 GB microSDXC card.[53] Kingmax released a comparable product in 2011.[54]

In late 2012, Lexar released the first 256 GB SDXC card, based on 20 nm NAND flash technology.[55]

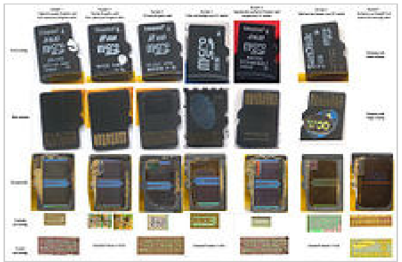
Markets

Secure Digital cards are used in many consumer electronic devices, and have become a widespread means of storing several gigabytes of data in a small size. Devices where the user may remove and replace cards often, such as digital cameras, camcorders, and video game consoles, tend to use full-sized cards. Devices where small size is paramount, such as mobile phones, tend to use microSD cards. SD cards are not the most economical solution in devices that need only a small amount of non-volatile memory, such as station presets in small radios. They may also not present the best choice for applications where higher storage capacities or speeds are a requirement as provided by other flash card standards such as CompactFlash.

Many personal computers of all types and personal digital assistants (PDAs) use SD cards, either through built-in slots or through an active electronic adaptor. Adaptors exist for the PC card, ExpressBus, USB, FireWire, and the parallel printer port. Active adaptors also let SD cards be used in devices designed for other formats, such as CompactFlash. The FlashPath adaptor lets SD cards be used in a floppy disk drive.

Fakes

Commonly found on the market are mislabeled or counterfeit Secure Digital cards that report a fake capacity or run slower than labeled.[56][57][58]



Images of genuine, questionable, and fake/counterfeit microSD (Secure Digital) cards before and after decapsulation. Details at source.

Digital cameras

SD/MMC cards replaced Toshiba's SmartMedia as the dominant memory card format used in digital cameras. In 2001, SmartMedia had achieved nearly 50% use, but by 2005 SD/MMC had achieved over 40% of the digital camera market and SmartMedia's share had plummeted by 2007.

At this time all the leading digital camera manufacturers used SD in their consumer product lines, including Canon, Casio, Fujifilm, Kodak, Leica, Nikon, Olympus, Panasonic, Pentax, Ricoh, Samsung, and Sony. Formerly, Olympus and Fujifilm used XD-Picture Cards (xD cards) exclusively, while Sony only used Memory Stick; by early 2010 all three supported SD.

Some prosumer and professional digital cameras continued to offer CompactFlash (CF), either on a second card slot or as the only storage, as CF supports much higher maximum capacities and faster transfer speeds and historically was cheaper for the same capacity.

Secure Digital memory cards can be used in Sony XDCAM EX camcorders with an adapter.[59]

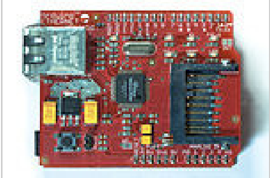
Personal computers

Although many personal computers accommodate SD cards as an auxiliary storage device through a built-in slot or a USB adaptor, SD cards cannot be used as the primary hard disk through the onboard ATA controller because none of the SD card variants supports ATA signalling. This use requires a separate SD controller chip[60] or a SD-to-CompactFlash converter. However, on computers that support bootstrapping from a USB interface, an SD card in a USB adaptor can be the primary hard disk, provided it contains an operating system that supports USB access once the bootstrap is complete.

Since late 2009, newer Apple computers with installed SD card readers have been able to boot in Mac OS X from SD storage devices, when properly formatted to Mac OS Extended file format and the default partition table set to GUID.[61] (See Other file systems below.)

Problems with using an SD card as the primary hard disk include the speed of the card, and that intensive writes to the card reduce its rated lifespan.[*citation needed*]

Embedded systems



A shield (daughterboard) that gives Arduino prototyping microprocessors access to SD cards.

In 2008, the SDA specified Embedded SD, "leverag[ing] well-known SD standards" to enable non-removable SD-style devices on printed circuit boards.[62] SanDisk provides such memory components under the iNAND brand.[63]

Most modern microcontrollers have built-in SPI logic that can interface to an SD card operating in its SPI mode, providing non-volatile storage. Even if a microcontroller lacks the SPI feature, the feature can be emulated by bit banging. For example, a home-brew hack combines spare General Purpose Input/Output (GPIO) pins of the processor of the Linksys WRT54G router with MMC support code from the Linux kernel.[64] This technique can achieve throughput of up to 1.6 Mbit/s.

Technical details

Physical size

The SD card specification defines three physical sizes. The SD and SDHC families are available in all three sizes, but the SDXC family is not available in the mini size, and the SDIO family is not available in the micro size.

Smaller cards are usable in larger slots through use of a passive adapter. By comparison, Reduced Size MultiMediaCards (RS-MMCs) are simply shorter MMCs and can be used in MMC slots by use of a physical extender.

Standard size

Size comparison of families: SD (blue), miniSD (green), microSD (red)

- SD (SDSC), SDHC, SDXC, SDIO
- 32.0×24.0×2.1 mm (1.26×0.94×0.083 in)
- 32.0×24.0×1.4 mm (1.26×0.94×0.055 in) (as thin as MMC) for **Thin SD** (rare)

Mini size

- miniSD, miniSDHC, miniSDIO
- 21.5×20.0×1.4 mm (0.85×0.79×0.055 in)

Micro size

The micro form factor is the smallest SD card format.[65]

- microSD, microSDHC, microSDXC
- 15.0×11.0×1.0 mm (0.59×0.43×0.039 in)

Transfer modes

Cards may support various combinations of the following bus types and transfer modes. The SPI bus mode and one-bit SD bus mode are mandatory for all SD families, as explained in the next section. Once the host device and the SD card negotiate a bus interface mode, the usage of the numbered pins is the same for all card sizes.

- **SPI bus mode:** Serial Peripheral Interface Bus is primarily used by embedded microcontrollers. This bus type supports only a 3.3-volt interface. This is the only bus type that doesn't require a host license.
- **One-bit SD bus mode:** Separate command and data channels and a proprietary transfer format.
- **Four-bit SD bus mode:** Uses extra pins plus some reassigned pins. UHS-I and UHS-II requires this bus type.

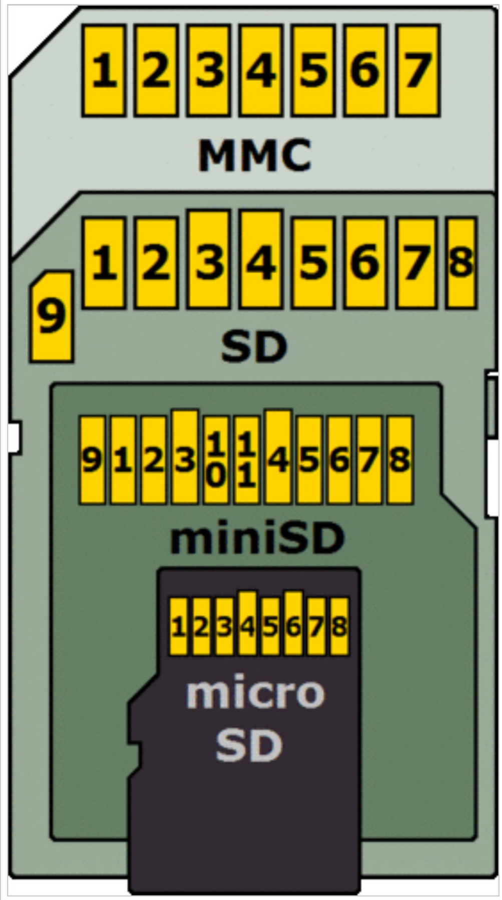
The physical interface comprises 9 pins, except that the miniSD card adds two unconnected pins in the center and the microSD card omits one of the two V_{SS} (Ground) pins.[66]

SPI Bus Mode

MMC Pin	SD Pin	miniSD Pin	microSD Pin	Name	I/O	Logic	Description
1	1	1	2	nCS	I	PP	SPI Card Select [CS] (Negative Logic)
2	2	2	3	DI	I	PP	SPI Serial Data In [MOSI]
3	3	3		VSS	S	S	Ground
4	4	4	4	VDD	S	S	Power
5	5	5	5	CLK	I	PP	SPI Serial Clock [SCLK]
6	6	6	6	VSS	S	S	Ground
7	7	7	7	DO	O	PP	SPI Serial Data Out [MISO]
	8	8	8	NC nIRQ	. O	. OD	Unused (memory cards) Interrupt (SDIO cards) (Negative Logic)
	9	9	1	NC	.	.	Unused
		10		NC	.	.	Reserved
		11		NC	.	.	Reserved

One-Bit SD Bus Mode

MMC Pin	SD Pin	miniSD Pin	microSD Pin	Name	I/O	Logic	Description
1	1	1	2	NC	.	.	Unused
2	2	2	3	CMD	I/O	PP, OD	Command, Response
3	3	3		VSS	S	S	Ground
4	4	4	4	VDD	S	S	Power
5	5	5	5	CLK	I	PP	Serial Clock
6	6	6	6	VSS	S	S	Ground
7	7	7	7	DAT0	I/O	PP	SD Serial Data 0
	8	8	8	NC nIRQ	. O	. OD	Unused (memory cards) Interrupt (SDIO cards) (Negative Logic)
	9	9	1	NC	.	.	Unused
		10		NC	.	.	Reserved
		11		NC	.	.	Reserved



Official pin numbers for each card type (top to bottom): MMC, SD, miniSD, microSD. This shows the evolution from the older MMC, on which SD is based.

Four-Bit SD Bus Mode

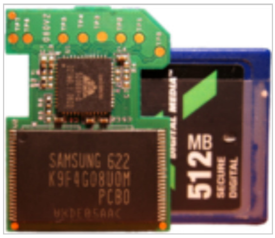
MMC Pin	SD Pin	miniSD Pin	microSD Pin	Name	I/O	Logic	Description
.	1	1	2	DAT3	I/O	PP	SD Serial Data 3
.	2	2	3	CMD	I/O	PP, OD	Command, Response
.	3	3		VSS	S	S	Ground
.	4	4	4	VDD	S	S	Power
.	5	5	5	CLK	I	PP	Serial Clock
.	6	6	6	VSS	S	S	Ground
.	7	7	7	DAT0	I/O	PP	SD Serial Data 0
	8	8	8	DAT1 nIRQ	I/O O	PP OD	SD Serial Data 1 (memory cards) Interrupt Period (SDIO cards share pin via protocol)
	9	9	1	DAT2	I/O	PP	SD Serial Data 2
		10		NC	.	.	Reserved
		11		NC	.	.	Reserved

Notes:

1. Direction is relative to card. I = Input, O = Output.
2. PP = Push-Pull logic, OD = Open-Drain logic.
3. S = Power Supply, NC = Not Connected (or logical high).

Interface

Command interface



Inside a 512 MB SD card: NAND flash chip that holds the data (bottom) and SD controller (top)

SD cards and host devices initially communicate through a synchronous one-bit interface, where the host device provides a clock signal that strobes single bits in and out of the SD card. The host device thereby sends 48-bit commands and receives responses. The card can signal that a response will be delayed, but the host device can abort the dialogue.

Through issuing various commands, the host device can:

- Determine the type, memory capacity, and capabilities of the SD card
- Command the card to use a different voltage, different clock speed, or advanced electrical interface
- Prepare the card to receive a block to write to the flash memory, or read and reply with the contents of a specified block.

The command interface is an extension of the MultiMediaCard (MMC) interface. SD cards dropped support for some of the commands in the MMC protocol, but added commands related to copy protection. By using only commands supported by both standards until determining the type of card inserted, a host device can accommodate both SD and MMC cards.

Inside a 2 GB SD card: two NAND flash chips (top and middle), SD controller chip (bottom)

Electrical interface

All SD card families initially use a 3.3-volt electrical interface. On command, SDHC and SDXC cards switch to 1.8-volt operation.[18]

At initial power-up or card insertion, the host device selects either the Serial Peripheral Interface (SPI) bus or the one-bit SD bus by the voltage level present on Pin 1. Thereafter, the host device may issue a command to switch to the four-bit SD bus interface, if the SD card supports it. For various card types, support for the four-bit SD bus is either optional or mandatory.[18]

After determining that the SD card supports it, the host device can also command the SD card to switch to a higher transfer speed. Until determining the card's capabilities, the host device should not use a clock speed faster than 400 kHz. SD cards other than SDIO (see below) have a "Default Speed" clock rate of 25 MHz. The host device is not required to use the maximum clock speed that the card supports. It may operate at less than the maximum clock speed to conserve power.[18] Between commands, the host device can stop the clock entirely.

SDIO cards

Inside a 16 GB SDHC card

The SDIO family comprises Low-Speed and Full-Speed cards. Both types of SDIO cards support SPI and one-bit SD bus types. Low-Speed SDIO cards are allowed to also support the four-bit SD bus; Full-Speed SDIO cards are required to support the four-bit SD bus. To use a SDIO card as a "combo card" (for both memory and I/O), the host device must first select four-bit SD bus operation. Two other unique features of Low-Speed SDIO are a maximum clock rate of 400 kHz for all communications, and the use of Pin 8 as "interrupt" to try to initiate dialogue with the host device.[67]

Ganging cards together

The one-bit SD protocol was derived from the MMC protocol, which envisioned the ability to put up to 3 cards on a bus of common signal lines. The cards use open collector interfaces, where a card may pull a line to the low voltage level; the line is at the high voltage level (because of a pull-up resistor) if no card pulls it low. Though the cards shared clock and signal lines, each card had its own chip select line to sense that the host device had selected it.

The SD protocol envisioned the ability to gang 30 cards together without separate chip select lines. The host device would broadcast commands to all cards and identify the card to respond to the command using its unique serial number.

In practice, cards are rarely ganged together because open-collector operation has problems at high speeds and increases power consumption. Newer versions of the SD specification recommend separate lines to each card.

Achieving higher card speeds

The SD specification defines four-bit-wide transfers. (The MMC specification supports this and also defines an eight-bit-wide mode.) Transferring several bits on each clock pulse improves the card speed. Advanced SD families have also improved speed by offering faster clock frequencies and double data rate (explained here).

File system

Like other types of flash memory card, an SD card of any SD family is a block-addressable storage device, in which the host device can read or write fixed-size blocks by specifying their block number.

MBR and FAT

Most SD cards ship preformatted with one or more MBR partitions, where the first or only partition contains a file system. This lets them operate like the hard disk of a personal computer. Per the SD card specification, an SD card is formatted with MBR and the following file system:

- For SDSC cards:
  - Capacity of less than 32680 logical sectors (smaller than 16 MB): FAT12 with partition type 01h and BPB 3.0 or EBPB 4.1[68]
  - Capacity of 32680 to 65535 logical sectors (between 16 MB and 32 MB): FAT16 with partition type 04h and BPB 3.0 or EBPB 4.1[68]
  - Capacity of at least 65536 logical sectors (larger than 32 MB): FAT16B with partition type 06h and EBPB 4.1[68]
- For SDHC cards:
  - Capacity of less than 16450560 logical sectors (smaller than 7.8 GB): FAT32 with partition type 0Bh and EBPB 7.1
  - Capacity of at least 16450560 logical sectors (larger than 7.8 GB): FAT32X with partition type 0Ch and EBPB 7.1
- For SDXC cards: exFAT with partition type 07h

Most consumer products that take an SD card expect that it is partitioned and formatted in this way. Universal support for FAT12, FAT16, FAT16B, FAT32 and FAT32X allows the use of SDSC and SDHC cards on most host computers with a compatible SD reader, to present the user with the familiar method of named files in a hierarchical directory tree.

On such SD cards, standard utility programs such as Mac OS X's "Disk Utility" or Windows' SCANDISK can be used to repair a corrupted filing system and sometimes recover deleted files. Defragmentation tools for FAT file systems may be used on such cards. The resulting consolidation of files may provide a marginal improvement in the time required to read or write the file,[69] but not an improvement comparable to defragmentation of hard drives, where storing a file in multiple fragments requires additional physical, and relatively slow, movement of a drive head. Moreover, defragmentation performs writes to the SD card that count against the card's rated lifespan. The write endurance of the physical memory is discussed in the article on flash memory; newer technology to increase the storage capacity of a card provides worse write endurance.

When reformatting an SD card with a capacity of at least 32 MB (65536 logical sectors or more), but not more than 2 GB, FAT16B with partition type 06h and EBPB 4.1[68] is recommended if the card is for a consumer device. (FAT16B is also an option for 4 GB cards, but it requires the use of 64 kiB clusters, which are not widely supported.) FAT16B does not support cards above 4 GB at all.

The SDXC specification mandates the use of Microsoft's proprietary exFAT file system,[70] which is supported only by some proprietary operating systems.

Other file systems

Because the host views the SD card as a block storage device, the card does not require MBR partitions or any specific file system. The card can be reformatted to use any file system the operating system supports. For example:

- Under Windows, SD cards can be formatted using NTFS and, on later versions, exFAT.
- Under Mac OS X, SD cards can be partitioned as GUID devices and formatted with the HFS Plus file system or still use exFAT.
- Under Unix-like operating systems such as Linux or FreeBSD, SD cards can be formatted using the UFS, Ext2, Ext3, Ext4, btrfs, HFS Plus, ReiserFS or F2FS file system.

Any recent version of the above can format SD cards using the UDF file system.

Additionally, as with live USB flash drives, an SD card can have an operating system installed on it. Computers that can boot from an SD card (either using a USB adapter or inserted into the computer's flash media reader) instead of the hard disk drive may thereby be able to recover from a corrupted hard disk drive.[*citation needed*] Such an SD card can be write-locked to preserve the system's integrity.

Risks of reformatting

Reformatting an SD card with a different file system, or even with the same one, may make the card slower, or shorten its lifespan. Some cards use wear leveling, in which frequently modified blocks are mapped to different portions of memory at different times, and some wear-leveling algorithms are designed for the access patterns typical of the file allocation table on a FAT12, FAT16 or FAT32 device.[71] In addition, the preformatted file system may use a cluster size that matches the erase region of the physical memory on the card; reformatting may change the cluster size and make writes less efficient.

SD/SDHC/SDXC memory cards have a "Protected Area" on the card for the SD standard's security function; a standard formatter may erase it, causing problems if security is used. The SD Association provides free SD Formatter software to overcome these problems.[72] The SD Formatter does not format the "Protected Area", and the Association recommends the use of appropriate application software or SD-compatible device that provides SD security function to format the "Protected Area" in the memory card.

Power consumption

The power consumption of SD cards varies by its speed mode, manufacturer and model.

During transfer it may be in the range of 66–330 mW (20–100 mA at a supply voltage of 3.3 V). Specifications from TwinMos technologies list a maximum of 149 mW (45 mA) during transfer. Toshiba lists 264–330 mW (80–100 mA).[73] Standby current is much lower, less than 0.2 mA for one 2006 microSD card.[74] If there is data transfer for significant periods, battery life may be reduced noticeably (smartphones typically have batteries of capacity around 6 Wh (Samsung Galaxy S2, 1650 mAh @ 3.7 V).

Modern UHS-II cards can consume up to 2.88 W, if the host device support bus speed mode SDR104 or UHS-II. Minimum power consumption in case of an UHS-II host is 0.72 W.

Card requirements regarding bus speed modes[75]

Bus Speed Mode *1	Max. Bus Speed [MB/s]	Max. Clock Frequency [MHz]	Signal Voltage [V]	SDSC [W]	SDHC [W]	SDXC [W]
HD312	312	52	0.4	-	2.88	2.88
FD156	156	52	0.4	-	2.88	2.88
SDR104	104	208	1.8	-	2.88	2.88
SDR50	50	100	1.8	-	1.44	1.44
DDR50	50	50	1.8	-	1.44	1.44
SDR25	25	50	1.8	-	0.72	0.72
SDR12	12.5	25	1.8	-	0.36	0.36 / 0.54
High Speed	25	50	3.3	0.72	0.72	0.72
Default Speed	12.5	25	3.3	0.33	0.36	0.36 / 0.54

Storage capacity and incompatibilities

All SD cards let the host device determine how much information the card can hold, and the specification of each SD family gives the host device a guarantee of the maximum capacity a compliant card reports.

By the time the Version 2.0 (SDHC) specification was completed in June 2006,[76] vendors had already devised 2 GB and 4 GB SD cards, either as specified in Version 1.01, or by creatively reading Version 1.00. The resulting cards do not work correctly in some host devices.[77][78]

SDSC cards above 1 GB

A host device can ask any inserted SD card for its 128-bit identification string (the Card-Specific Data or CSD). In standard-capacity cards (SDSC), 12 bits identify the number of memory clusters (ranging from 1 to 4,096) and 3 bits identify the number of blocks per cluster (which decode to 4, 8, 16, 32, 64, 128, 256, or 512 blocks per cluster). The host device multiplies these figures (as shown in the following section) with the number of bytes per block to determine the card's capacity in bytes.

4 GB SDSC card

SD version 1.00 assumed 512 bytes per block. This permitted SDSC cards up to 4,096 × 512 × 512 = 1 GB, for which there are no known incompatibilities.

Version 1.01 let an SDSC card use a 4-bit field to indicate 1,024 or 2,048 bytes per block instead.[18] Doing so enabled cards with 2 GB and 4 GB capacity, such as the Transcend 4 GB SD card and the Memorette 4 GB SD card.

Early SDSC host devices that assume 512-byte blocks therefore do not fully support the insertion of 2 GB or 4 GB cards. In some cases, the host device can read data that happens to reside in the first 1 GB of the card. If the assumption is made in the driver software, success may be version-dependent. In addition, any host device might not support a 4 GB SDSC card, since the specification lets it assume that 2 GB is the maximum for these cards.

Storage capacity calculations

The format of the Card-Specific Data (CSD) register changed between version 1 (SDSC) and version 2.0 (which defines SDHC and SDXC).

Version 1

In Version 1 of the SD specification, capacities up to 2 GB are calculated by combining fields of the CSD as follows:

```
Capacity=(C_SIZE+1) × 2^(C_SIZE_MULT + 2 + READ_BL_LEN)
Where: 0 <= C_SIZE <= 4095,
0 <= C_SIZE_MULT <= 7,
READ_BL_LEN is 9 (for 512 bytes/sector) or 10 (for 1024 bytes/sector)
```

Later versions state (at Section 4.3.2) that a 2 GB SDSC card shall set its READ_BL_LEN (and WRITE_BL_LEN) to indicate 1024 bytes, so that the above computation correctly reports the card's capacity; but that, for consistency, the host device shall not request (by CMD16) block lengths over 512 bytes.[18]

Versions 2 and 3

In the definition of SDHC cards in Version 2.0, the C_SIZE portion of the CSD is 22 bits and it indicates the memory size in multiples of 512 KB (the C_SIZE_MULT field is removed and READ_BL_LEN is no longer used to compute capacity.) Two bits that were formerly reserved now identify the card family: 0 is SDSC; 1 is SDHC or SDXC; 2 and 3 are reserved.[18] Because of these redefinitions, older host devices do not correctly identify SDHC or SDXC cards nor their correct capacity.

- SDHC cards are restricted to reporting a capacity not over 32 GiB.
- SDXC cards are allowed to use all 22 bits of the C_SIZE field. An SDHC card that did so (reported C_SIZE > 65375 to indicate a capacity of over 32 GiB) would violate the specification. A host device that relied on C_SIZE rather than the specification to determine the card's maximum capacity might support such a card, but the card might fail in other SDHC-compatible host devices.

Capacity is calculated thus:

```
Capacity=(C_SIZE+1)*524288
where for SDHC  4112<=C_SIZE<=65375    (approx. 2 GB) < capacity < 32 GiB
for SDXC 65535<=C_SIZE 32 GiB <= capacity <= 2 TiB max.
```

Capacities above 4 GB can only be achieved by following Version 2.0 or later versions. In addition, capacities equal to 4 GB must also do so to guarantee compatibility.

Openness of specification

Like most memory card formats, SD is covered by numerous patents and trademarks. Royalties for SD card licences are imposed for manufacture and sale of memory cards and host adapters (US$1,000/year plus membership at US$1,500/year), but SDIO cards can be made without royalties.

Dismantled microSD to SD adapter showing the passive connection from the microSD card slot on the bottom to the SD pins on the top

Early versions of the SD specification were available only after agreeing to a non-disclosure agreement (NDA) that prohibited development of an open source driver. However, the system was eventually reverse-engineered, and free software drivers provided access to SD cards that did not use DRM. Since then, the (SDA) has provided a simplified version of the specification under a less restrictive license.[79] Although most open-source drivers were written before this, it has helped to solve compatibility issues.

In 2006, the SDA released a simplified version of the specification of the host controller interface (as opposed to the specification of SD cards) and later also for the physical layer, ASSD extensions, SDIO, and SDIO Bluetooth Type-A, under a disclaimers agreement.[80] Again, most of the information had already been discovered and Linux had a fully free driver for it. Still, building a chip conforming to this specification caused the One Laptop per Child project to claim "the first truly Open Source SD implementation, with no need to obtain an SDI license or sign NDAs to create SD drivers or applications."[81]

The proprietary nature of the complete SD specification affects embedded systems, laptop computers, and some desktop computers; many desktop computers do not have card slots, instead using USB-based card readers if necessary. These card readers present a standard USB mass storage interface to memory cards, thus separating the operating system from the details of the underlying SD interface. However, embedded systems (such as portable music players) usually gain direct access to SD cards and thus need complete programming information. Desktop card readers are themselves embedded systems; their manufacturers have usually paid the SDA for complete access to the SD specifications. Many notebook computers now include SD card readers not based on USB; device drivers for these essentially gain direct access to the SD card, as do embedded systems.

The SPI-bus interface mode is the only type that doesn't require a host license for accessing SD cards.

Comparison to other flash memory formats

Overall, SD is less open than CompactFlash or USB flash memory drives. Those open standards can be implemented without paying for licensing, royalties, or documentation. (CompactFlash and USB flash drives may require licensing fees for the use of the SDA's trademarked logos.)

However, SD is much more open than Memory Stick, for which no public documentation nor any documented legacy implementation is available. All SD cards can be accessed freely using the well-documented SPI bus.

Size comparison of various flash cards: SD, CompactFlash, MMC, xD

xD cards are simply 18-pin NAND flash chips in a special package and support the standard command set for raw NAND flash access. Although the raw hardware interface to xD cards is well understood, the layout of its memory contents—necessary for interoperability with xD card readers and digital cameras—is totally undocumented. The consortium that licenses xD cards has not released any technical information to the public.

Comparison of technical features of MMC and SD card variants

Type	MMC	RS-MMC	MMCplus	SecureMMC	SDIO	SD	miniSD	microSD
SD Socket	Yes	Extender	Yes	Yes	Yes	Yes	Adapter	Adapter
Pins	7	7	13	7	9	9	11	8
Width	24 mm	24 mm	24 mm	24 mm	24 mm	24 mm	20 mm	11 mm
Length	32 mm	18 mm	32 mm	32 mm	32 mm+	32 mm	21.5 mm	15 mm
Thickness	1.4 mm	1.4 mm	1.4 mm	1.4 mm	2.1 mm	2.1 mm (most) 1.4 mm (rare)	1.4 mm	1 mm
SPI bus mode	Optional	Optional	Optional	Yes	Yes	Yes	Yes	Yes
1-bit bus mode	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
4-bit bus mode	No	No	Yes	No	Optional	Optional	Optional	Optional
8-bit bus mode	No	No	Yes	No	No	No	No	No
Interrupts	No	No	No	No	Optional	No	No	No
Max clock rate	20 MHz	20 MHz	52 MHz	20 MHz?	50 MHz	208 MHz	208 MHz	208 MHz
Max transfer rate	20 Mbit/s	20 Mbit/s	416 Mbit/s	20 Mbit/s?	200 Mbit/s	832 Mbit/s	832 Mbit/s	832 Mbit/s
Max SPI transfer	20 Mbit/s	20 Mbit/s	52 Mbit/s	20 Mbit/s	50 Mbit/s	25 Mbit/s	50 Mbit/s	50 Mbit/s
DRM	No	No	No	Yes	N/A	Yes	Yes	Yes
User encrypt	No	No	No	Yes	No	No	No	No
Simplified spec	Yes	Yes	No	Not yet?	Yes	Yes	No	No
Membership cost	JEDEC: US$4,400/yr, optional				SD Card Association: US$2,000/year, general; US$4,500/year, executive			
Specification cost	Free			?	Simplified spec: Free. Full spec: Free to members, US$1,000/year to R&D non-members			
Host license	No	No	No	No	Yes, US$1,000/year. No, if using SPI-mode only.			
Card royalties	Yes	Yes	Yes	Yes	Yes, US$1,000/year	Yes	Yes	Yes
Open source compatible	Yes	Yes	Yes?	Yes?	Yes	Yes	Yes	Yes
Nominal operating voltage(s)	3.3 V	1.8 V/3.3 V	1.8 V/3.3 V[82][83]	1.8 V/3.3 V	3.3 V	3.3 V (SDSC), 1.8/3.3 V (SDHC), 1.8/3.3 V (SDXC)	3.3 V (miniSD), 1.8/3.3 V (miniSDHC)	3.3 V (SDSC), 1.8/3.3 V (microSDHC), 1.8/3.3 V (microSDXC)
Maximum available capacity	128 GB	2 GB	128 GB?	128 GB?	?	4 GB (SD), 32 GB (SDHC), 256 GB (SDXC)	4 GB (miniSD), 16 GB (miniSDHC)	4 GB (microSD), 32 GB (microSDHC), 128 GB (microSDXC)
Type	MMC	RS-MMC	MMCplus	SecureMMC	SDIO	SD	miniSD	microSD

- Table data compiled from MMC, SD, and SDIO specifications from SD Association and JEDEC web sites. Data for other card variations are interpolated.

See also

- Comparison of memory cards
- Serial Peripheral Interface Bus (SPI)
- Flash memory
- Universal Flash Storage

References

1. ^ *a b* "About" (http://www.sdcard.org/developers/about/). SD Card Association. Retrieved 2011-12-08.
2. ^ *a b* "Capacity" (http://www.sdcard.org/developers/overview/capacity/). SD Card Association. Retrieved 2011-12-08.
3. ^ *a b* "Using SDXC" (http://www.sdcard.org/consumers/sdxc_capabilities/using_sdxc/). SD Card Association. Retrieved 2011-12-08.
4. ^ "SDIO" (http://www.sdcard.org/developers/overview/sdio/). SD Card Association. Retrieved 2011-12-08.
5. ^ "Using SD Memory Cards is Easy" (video on YouTube) – SD Card Association (http://www.youtube.com/watch?v=wX6NwBa1csY)
6. ^ What are SDHC, miniSDHC, and microSDHC? (http://www.sandisk.com/Assets/File/pdf/retail/SDHC1.pdf) SanDisk.com
7. ^ What's new in Firmware 2.41 Beta (for COWON D2) (http://www.jetaudio.com/download/cowon_rn_d2.html#241) JetAudio
8. ^ "Microsoft Support 934428 – Hotfix for Windows XP that adds support for SDHC cards that have a capacity of more than 4 GB" (http://support.microsoft.com/kb/934428). Support.microsoft.com. 2008-02-15. Retrieved 2010-08-22.
9. ^ "Microsoft Support 939772 – Some Secure Digital (SD) cards may not be recognized in Windows Vista" (http://support.microsoft.com/kb/939772). Support.microsoft.com. 2008-05-15. Retrieved 2010-08-22.
10. ^ "Microsoft Support 949126 – A Secure Digital High Capacity (SDHC) card is not recognized on a Windows Vista Service Pack 1-based computer" (http://support.microsoft.com/kb/949126). Support.microsoft.com. 2008-02-21. Retrieved 2010-08-22.
11. ^ Exfat for BSD and Linux systems from Google Code (http://code.google.com/p/exfat/)
12. ^ WindowsXP-KB955704-x86-ENU.exe (2009-01-26). "Download: Update for Windows XP (KB955704) – Microsoft Download Center – Download Details" (http://www.microsoft.com/downloads/details.aspx?FamilyID=1cbe3906-ddd1-4ca2-b727-c2dff5e30f61&displaylang=en). Microsoft.com. Retrieved 2012-03-21.
13. ^ "About the SD and SDXC card slots" (http://support.apple.com/kb/HT3553). Apple Inc. 2011-05-03. Retrieved 2011-09-05.
14. ^ "Apple released exFAT support in OS X 10.6.5 update" (http://www.tuxera.com/mac/apple-released-exfat-support-in-os-x-10-6-5-update/). Tuxera.com. 2010-11-22. Retrieved 2012-01-04.
15. ^ https://www.sdcard.org/downloads/formatter_4/
16. ^ "Bus Speed (Default Speed/ High Speed/ UHS) - SD Association" (https://www.sdcard.org/developers/overview/bus_speed/). Sdcard.org. Retrieved 2013-11-13.
17. ^ "SD cards branded with an upper-case 'I' are faster, yo" (http://www.engadget.com/2010/06/24/sd-cards-branded-with-an-upper-case-i-are-faster-yo/). Engadget. 2010-06-24. Retrieved 2010-08-22.
18. ^ *a b c d e f g h i j* "SD Part 1, Physical Layer Simplified Specification, Version 3.01" (http://www.sdcard.org/downloads/pls/simplified_specs/archive/part1_301.pdf) (PDF). SD Association. 2010-05-18. Retrieved 2013-11-25.
19. ^ "SD Speed Class/UHS Speed Class" (https://www.sdcard.org/developers/overview/speed_class/). SD card. Retrieved 21 Nov 2011.
20. ^ "Association Triples Speeds with UHS-II" (https://www.sdcard.org/press/SD_Association_Announces_UHS-II_eBOOK_Jan_5_2011_ENGLISH.PDF). SD Card. 5 January 2011. Retrieved 2011-08-09.
21. ^ Galbraith, Rob (2011-01-05). "SD Card Association announces UHS-II, ultra high-speed SD card specification" (http://www.robgalbraith.com/bins/content_page.asp?cid=7-11133-11156). Retrieved 2011-01-05.
22. ^ "SD Speed Class - SD Association" (https://www.sdcard.org/developers/overview/speed_class). Sdcard.org. Retrieved 2013-11-13.
23. ^ About Compatibility with Host Devices (https://www.sdcard.org/consumers/compatibility/) SD Association
24. ^ SD Bus Speed (https://www.sdcard.org/consumers/speed/bus_speed/) SD Association
25. ^ *Speed Class* (https://www.sdcard.org/consumers/speed/), SDCard
26. ^ "NEW SDXC AND SDHC MEMORY CARDS SUPPORT 4K2K VIDEO" (https://www.sdcard.org/home/New_SDXC_and_SDHC_Memory_Cards_Now_Support_4K2K_Video_11-6-2013_FINAL_2.pdf). SD Association. Retrieved 2013-11-13.
27. ^ Kim, H; Agrawal, N; Ungureanu, C, *Revisiting Storage for Smartphones* (http://static.usenix.org/events/fast/tech/full_papers/Kim.pdf) (PDF), America: NEC Laboratories, table 3
28. ^ "Flash Memory Cards and X-Speed Ratings" (http://www.kingston.com/us/flash/x-speed). Kingston. Retrieved 2010-08-22.
29. ^ "Windows Phone 7 Secure Digital Card Limitations" (http://support.microsoft.com/kb/2450831).
30. ^ "Windows Phone 7's microSD mess: the full story (and how Nokia can help you out of it)" (http://www.engadget.com/2010/11/17/windows-phone-7s-microsd-mess-the-full-story-and-how-nokia-ca).
31. ^ "Super Talent Super Digital Overview & Specs" (http://reviews.cnet.com/secure-digital/super-talent-super-digital/4505-3236_7-32074490.html). Cnet.com. Retrieved 2012-07-28.
32. ^ "Eye-Fi" (http://www.eye.fi/). Retrieved 2010-08-22.
33. ^ AudioHolics (http://www.audioholics.com/news/industry-news/sandisk-slotmusic)
34. ^ "slotRadio" (http://www.sandisk.com/consumer-products/slotradio). SanDisk. Retrieved 2011-11-27.
35. ^ "SanDisk Ultra II SD Plus USB/SD card" (http://www.theregister.co.uk/2005/07/25/review_sandisk_ultra_ii_sd_plus/), *The Register* (UK), 2005-07-25
36. ^ "A-DATA Super Info SD Card 512MB" (http://www.techpowerup.com/reviews/AData/ADATASuperInfoSD/). Tech power up. 2007-02-20. Retrieved 2011-12-30.
37. ^ "Press Releases 17 July 2003" (http://www.toshiba.co.jp/about/press/2003_07/pr1701.htm). Toshiba. 2003-07-17. Retrieved 2010-08-22.
38. ^ SanDisk Introduces The World's Smallest Removable Flash Card For Mobile Phones-The miniSD Card (http://www.sandisk.com/Corporate/PressRoom/PressReleases/PressRelease.aspx?ID=1536) SanDisk.com

39. ^ PhoneScoop – Sandisk T-Flash announcement. (http://www.phonescoop.com/news/item.php?n=801)
40. ^ SanDisk Introduces 4 GB miniSDHC Flash Card for Mobile Phones (http://www.sandisk.com/Corporate/PressRoom/PressReleases/PressRelease.aspx?ID=3530) SanDisk.com
41. ^ "SanDisk and Sony to expand Memory Stick Pro and Memory Stick Micro formats" (http://www.sandisk.com/about-sandisk/press-room/press-releases/2009/2009-01-07-sandisk-and-sony-to-expand-%E2%80%9Cmemory-stick-pro%E2%80%9D-and-%E2%80%9Cmemory-stick-micro%E2%80%9D-formats). Sandisk. 2009-01-07. Retrieved 2010-08-22.
42. ^ "SD Card, Memory Stick formats to reach 2 terabytes, but when?" (http://www.betanews.com/article/SD_Card_Memory_Stick_formats_to_reach_2_terabytes_but_when/1231453659). *Beta news*. 2009-01-08.
43. ^ "Pretec introduces world's first SDXC card" (http://www.dpreview.com/news/0903/09030601pretecsdxc.asp). *Digital Photography Review*. 2009-03-06. Retrieved 2010-08-22.
44. ^ "Canon EOS Rebel T2i/550D Digital SLR Camera Review" (http://www.the-digital-picture.com/Press-Release/Canon-EOS-Rebel-T2i-550D-Digital-SLR-Camera-Press-Release.aspx). The Digital Picture.
45. ^ Ng, Jansen (2009-11-24). "Lack of Card Readers Holding Back SDXC Flash Memory Adoption" (http://www.dailytech.com/Lack+of+Card+Readers+Holding+Back+SDXC+Flash+Memory+Adoption/article16915.htm). DailyTech. Retrieved 2009-12-22.
46. ^ Ng, Jansen (2009-11-30). "Lenovo, HP, Dell Integrating SDXC Readers in New 32nm Intel "Arrandale" Laptops" (http://www.dailytech.com/Lenovo+HP+Dell+Integrating+SDXC+Readers+in+New+32nm+Intel+Arrandale+Laptops/article16937.htm). DailyTech. Retrieved 2009-12-22.
47. ^ Ng, Jansen (2009-12-22). "Toshiba Sampling First SDXC Flash Memory Cards" (http://www.dailytech.com/Toshiba+Sampling+First+SDXC+Flash+Memory+Cards/article16972.htm). DailyTech. Retrieved 2009-12-22.
48. ^ "Toshiba's 64 GB SDXC card to finally go on sale (in Japan)" (http://www.crunchgear.com/2010/04/15/toshibas-64gb-sdxc-card-to-finally-go-on-sale-in-japan/). CrunchGear. Retrieved 2010-08-09.
49. ^ "Panasonic Introduces New 64 GB* and 48 GB* SDXC Memory Cards, Available Globally in February 2010" (http://www2.panasonic.com/webapp/wcs/stores/servlet/prModelDetail?storeId=11301&catalogId=13251&itemId=389511&modelNo=Content01052010041118461&surfModel=Content01052010041118461). Panasonic. Retrieved 2010-08-09.
50. ^ "Sandisk ships its highest capacity sd card ever" (http://sandisk.com/about-sandisk/press-room/press-releases/2010/2010-02-22-sandisk-ships-its-highest-capacity-sd-card-ever,-the-64gb-sandisk-ultra-sdxc-card). SanDisk. 2010-02-22. Retrieved 2010-08-09.
51. ^ Lexar ships 128 GB Class 10 SDXC card; March 2011. (http://www.betanews.com/article/Lexar-ships-first-128GB-SDXC-cards/1300305310)
52. ^ "SDXC/SDHC 433X Class 16 Card from Pretec" (http://www.pretec.com/news-event/press-room/item/root/sdxcsdhc-433x-c16). Pretec. 2011-06-13. Retrieved 2010-12-03.
53. ^ *First 64GB microSD Card Here; When Will Smartphones Support It?* (http://pocketnow.com/smartphone-news/first-64gb-microsd-card-arrives-when-will-smartphones-support), "Smartphone news", *Pocket now*
54. ^ "Kingmax flaunts world's first 64GB microSD card" (http://www.engadget.com/2011/05/26/kingmax-flaunts-worlds-first-64gb-microsd-card/), *Engadget*, 2011-05-26
55. ^ "Lexar Announces Industry's First 256GB SDXC UHS-I Memory Card" (http://www.lexar.com/about/newsroom/press-releases/lexar-announces-industry-s-first-256gb-sdxc-uhs-i-memory-card). Micron. Retrieved 22 December 2012.
56. ^ Andrew (bunnie) Huang (of chumby). "On MicroSD Problems" (http://www.bunniestudios.com/blog/?page_id=1022). Retrieved 29 May 2013.
57. ^ Schnurer, Georg (2007-02-28). "Gefälschte SD-Karten" (http://www.heise.de/mobil/meldung/Gefaelschte-SD-Karten-151283.html) (in German). Heise mobile – c't magazin für computertechnik. Retrieved 2013-06-07.
58. ^ Feddern, Boi (2013-03-18). "Smartphones wählerisch bei microSDHC-Karten" (http://www.heise.de/mobil/meldung/Smartphones-waehlerisch-bei-microSDHC-Karten-1825259.html) (in German). Heise mobile – c't magazin für computertechnik. Retrieved 2013-06-09.
59. ^ MEAD-SD01 SDHC card adapter (Sony) (http://pro.sony.com/bbsc/ssr/micro-xdcamexsite/cat-accessories/product-MEADSD01/)
60. ^ "TS-7800 Embedded" (http://www.embeddedarm.com/products/board-detail.php?product=TS-7800). Embeddedarm.com. Retrieved 2010-08-22.
61. ^ "About the SD and SDXC card slot" (http://support.apple.com/kb/HT3553). Support.apple.com. 2013-06-08. Retrieved 2013-11-13.
62. ^ "Embedded SD" (http://www.sdcard.org/developers/overview/embedded_sd/). SD Card Association. Retrieved 2011-11-30.
63. ^ "iNAND Embedded Flash Drives" (http://www.sandisk.com/business-solutions/inand-embedded-flash-drives). SanDisk. Retrieved 2011-11-30.
64. ^ "Linksys WRT54G-TM SD/MMC mod – DD-WRT Wiki" (http://www.dd-wrt.com/wiki/index.php/Linksys_WRT54G-TM_SD/MMC_mod). Dd-wrt.com. 2010-02-22. Retrieved 2010-08-22.
65. ^ "About" (https://www.sdcard.org/consumers/choices/). SD Card Association. Retrieved 2011-05-02.
66. ^ SD Part 1, Physical Layer Simplified Specification, Version 4.10 (https://www.sdcard.org/downloads/pls/simplified_specs/part1_410.pdf)
67. ^ "Simplified Version of SDIO CARD SPEC" (https://www.sdcard.org/developers/overview/sdio/sdio_spec/). SD Card Association. Retrieved 2011-12-09.
68. ^ ***a b c d*** *SD Memory Card Specifications – PART 2 FILE SYSTEM SPECIFICATION – Version 1.0*. 1.0. SD Group, Matsushita Electric Industrial Co., Ltd. (MEI), SanDisk Corporation, Toshiba Corporation. 2000-02.
69. ^ *Fragmentation and Speed* (https://www.sdcard.org/developers/overview/speed_class/), SD Card
70. ^ "SDXC memory cards promise 2 TB of storage, 300 MBps transfer" (http://www.engadget.com/2009/01/07/sdxc-memory-cards-promise-2tb-of-storage-300mbps-transfer/). Engadget. 2009-01-07. Retrieved 2010-08-22.
71. ^ "Optimizing Linux with cheap flash drives" (http://lwn.net/Articles/428584/). Linux Weekly News. Retrieved 2011-04-11.
72. ^ SD Formatter 3.1 for SD/SDHC/SDXC (https://www.sdcard.org/downloads/formatter_3/), SD Association
73. ^ "microSD & microSDHC Cards" (http://www.toshiba-memory.com/en/micro_sd_cards.html), *Memory Solutions*, Toshiba
74. ^ *Micro SD specification* (http://www.dtt8.com/images/micro-sd%20specification.pdf) (PDF), DTT
75. ^ "SD Specifications Version 4.10" (https://docs.google.com/viewer?url=https://www.sdcard.org/downloads/pls/simplified_specs/part1_410.pdf), *3.10.5 – Summary of Bus Speed Mode for UHS-II Card*, SD Card Association

76. ^ "A look into how SDHC will affect the future Nand Flash market" (http://www.dramexchange.com/WeeklyResearch/Post/1/492.aspx), *DRAMeXchange*, December 2006
77. ^ "SD Compatibility" (http://www.hjreggel.net/cardspeed/special-sd.html), *Card spped – Card Readers and Memory Cards*, HJ Reggel, December 1, 2006
78. ^ "WinXP SP3 can't read 4GB SD card in multicard reader" (http://www.eggheadcafe.com/software/aspnet/33338344/winxp-sp3-cant-read-4gb.aspx). Egg head cafe. Retrieved 2010-08-22.
79. ^ "Sharp Linux PDA promotes the use of proprietary SD card, but more open MMC works just fine" (http://www.linux.com/archive/feed/20060). Linux.com. Retrieved 2010-08-22.
80. ^ Simplified Specification Agreement (http://www.sdcard.org/developers/tech/sdcard/pls/) from the SDA's website
81. ^ "OLPC mailing list archive" (http://mailman.laptop.org/pipermail/community-news/2006-September/000023.html). Mailman.laptop.org. Retrieved 2010-08-22.
82. ^ *MMC 4.4 Standard* (http://www.jedec.org/download/search/JESD84-A44.pdf) (PDF), JEDEC, 2008, p. 7.
83. ^ *v4.0 card does not support 1.8V* (http://www.transcendusa.com/support/dlcenter/datasheet/TSxxMMC4.pdf) (PDF), US: Transcend, 2009.

External links

Official

- Official website (http://www.sdcard.org)
- Official SD simplified specification is Free (https://www.sdcard.org/downloads/pls/)
- Official SD full specification is US$1000 / year (https://www.sdcard.org/developers/licensing/)
- SD Formatter for SD / SDHC / SDXC cards (Windows and Mac) (https://www.sdcard.org/downloads/formatter_3/), SD card.

Interfacing

- *Interfacing to SD cards* (http://elm-chan.org/docs/mmc/mmc_e.html), Elm chan.
- *Interfacing AVR (Arduino) to SD cards, C source code* (http://www.dharmanitech.com/2009/01/sd-card-interfacing-with-atmega8-fat32.html), Dharmani tech.
- *Interfacing ARM to SD cards, C source code* (http://gandalf.arubi.uni-kl.de/avr_projects/arm_projects/arm_memcards/), DE: Uni KL.
- *Interfacing MSP430 to SD cards, C source code* (http://www.cs.ucr.edu/~amitra/sdcard/Additional/sdcard_appnote_foust.pdf) (PDF), Michigan State University.
- *Interfacing MAXQ2000 to SD cards, good technical descriptions, C source code* (http://www.maxim-ic.com/appnotes.cfm/an_pk/3969), Maxim IC.
- *SD card controller, Verilog source code* (http://www.opencores.org/project,sdcard_mass_storage_controller), Open cores.

Categories: 1999 introductions | Computer storage devices | Solid-state computer storage media

Elastic Sheep

Because elasticdog was already taken

- front page
- About
- Shop
- Archives
- RSS

← Installing the LUFA TeensyHID bootloader Reading an SD card part 2: Teensy 2.0 →

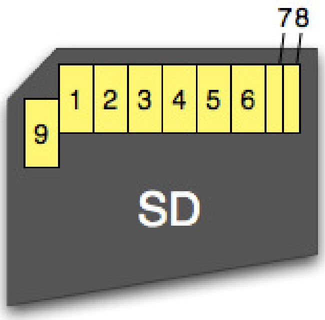
Reading an SD card with an ATMEGA168

January 12th, 2010 · 28 Comments · Uncategorized

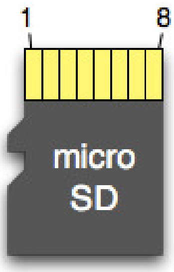
In this small week-end project, I will connect a SD card to an ATMEGA168 and try to read data from it.

The SD card pin-out

The SD card can be talked to with three different transfer modes: 1-bit SD mode, 4-bit SD mode and SPI mode. According to Wikipedia, all cards must support all three modes except for micro SD where the SPI mode is optional. I will nonetheless try to read my micro SD with the SPI mode.



Pin	SD	SPI
1	CD/DAT3	CS
2	CMD	DI
3	VSS1	VSS1
4	VDD	VDD
5	CLK	SCLK
6	VSS2	VSS2
7	DAT0	DO
8	DAT1	X
9	DAT2	X



Pin	SD	SPI
1	DAT2	X
2	CD/DAT3	CS
3	CMD	DI
4	VDD	VDD
5	CLK	SCLK
6	VSS	VSS
7	DAT0	DO
8	DAT1	X

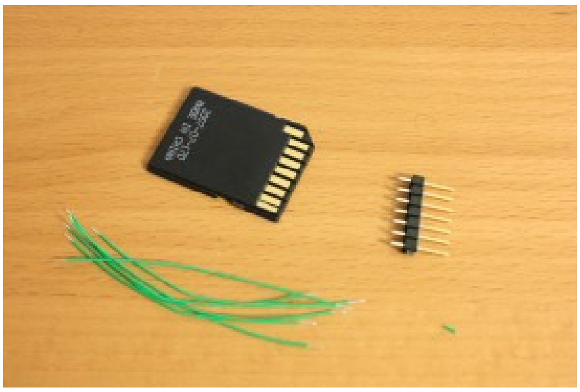
[**Edit** Correct pinout table added thanks to John Ulyate's comment]

Connecting the card to a circuit

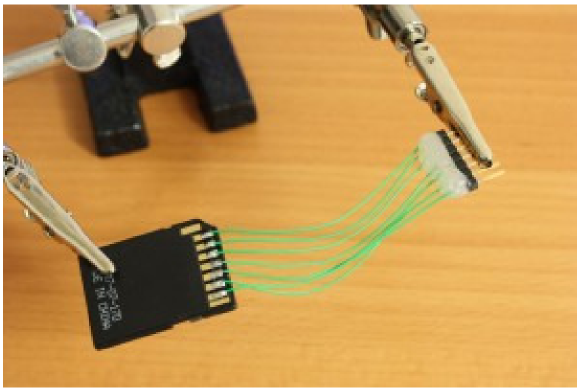
I found a circuit example on the following page http://www.captain.at/electronic-atmega-mmc.php. The MCU is a 5V-powered ATMEGA16. The SD card is powered through a 3.3V regulator. The author used resistor dividers to connect the SD card inputs to the ATMEGA16 SPI pins.

I bought the same regulator and thought, why not instead try to power the ATMEGA168 directly with 3.3V thus avoiding the need for a voltage adaptation. The ATMEGA168 datasheet tells me that at 3.3V the maximum safe frequency is 16MHz so I can keep my current crystal.

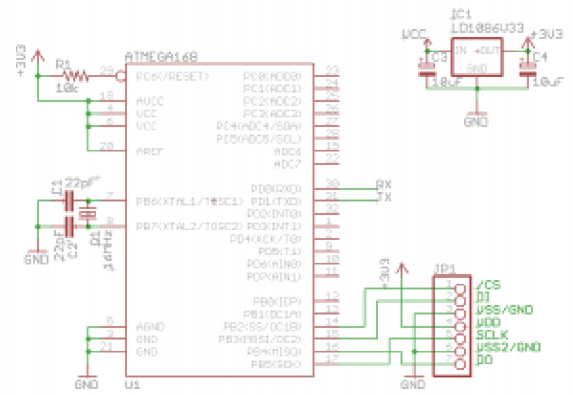
I improvised an interface for a micro SD with a SD card adapter and a 7-pins header.

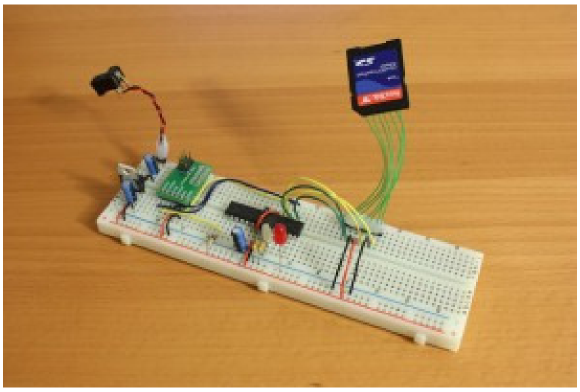


I used some thin wire to avoid stressing the SD Card pads and added some hot glue to have a stronger header connector.



Here is the schematic of my version of the circuit and a picture of the resulting breadboard:

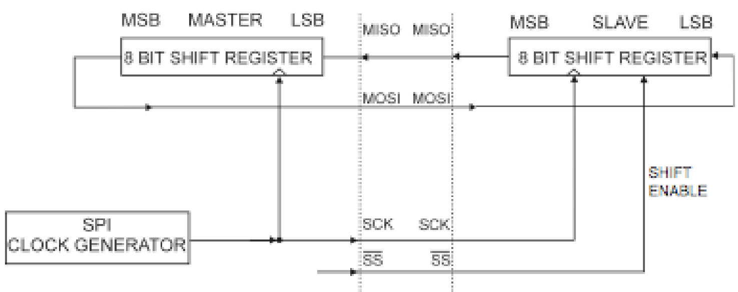




Understanding the SPI interface

SPI stands for Serial Peripheral Interface. This interface allows high speed synchronous transfers between a MCU and a peripheral or another MCU. The ATMEGA168 datasheet contains the following figure that highlights the powerful simplicity of this interface:

Figure 18-2. SPI Master-slave Interconnection



Each side of the interface has a 8-bit shift register. The left side is the master and the right side is the slave. When the master initiates a transfer to the slave, its SPI clock signal triggers a bit-by-bit copy of each register to the other one. Sending a byte from the master always involves receiving a byte from the slave.

The master SPI clock dictates the speed of the transfer. For the ATMEGA168 the maximum speed is Fclk/2. So with a 16 MHz crystal, the maximum theoretical bandwidth is 8Mbit/s, or 1MByte/s. A regular card should support a clock of 25 MHz.

Here is some code adapted from the ATMEGA168 datasheet to initialize the SPI interface and do a transfer. A single function is enough to send or receive a byte or both.

```
void spi_init(void)
{
  /* Set MOSI and SCK output */
  DDR_SPI |= _BV(DD_MOSI) | _BV(DD_SCK);
  DDR_SPI &= ~_BV(DD_MISO);

  /* Enable SPI, Master, set clock rate fck/128 */
  SPCR = _BV(SPE) | _BV(MSTR) | _BV(SPR0) | _BV(SPR1);
  SPSR = _BV(SPI2X);
}

uint8_t spi_send_receive_byte(uint8_t byte)
{
  /* Start transmission */
  SPDR = byte;

  /* Wait for transmission complete */
  while (!(SPSR & _BV(SPIF)));

  /* Read the received byte */
  byte = SPDR;

  return byte;
}
```

The interface can be tested with a loopback: by wiring the MOSI output to the MISO input, you can check that the received byte match the sent byte.

Implementing the MMC protocol

The protocol to use with the SPI interface is defined by Sandisk (ProdManualSDCardv1.9.pdf)

For a simple implementation, I started from the AN10406 LPC2000 application note (Accessing SD/MMC card using SPI on LPC2000). I adapted the mmc_init and mmc_read_block functions to be able to read the first block of a SD card.

The full program is downloadable at the end of the post. The serial port speed is set at 57600 bauds.

Here is the serial output of reading the sector 63 of my test SD card. It contains a FAT32 MBR with a descriptor for the only available partition "NO NAME":

```
spi_init
mmc_init
send CMD0
send dummy clocks
send CMD1
send dummy clocks
send CMD16
mmc_init end
read block 63
receive block
block received
0000: eb58904d53444f53352e300002082400 .X.MSDOS5.0...$.
0010: 0200000000f800003f00ff003f000000 ........?...?...
0020: 99933c001e0f00000000000002000000 ..<.............
0030: 01000600000000000000000000000000 ................
0040: 000029696561a84e4f204e414d452020 ..)iea.NO NAME
0050: 2020464154333220202033c98ed1bcf4   FAT32   3.....
0060: 7b8ec18ed9bd007c884e028a5640b408 {......|.N..V@..
0070: cd137305b9ffff8af1660fb6c640660f ..s......f...@f.
0080: b6d180e23ff7e286cdc0ed0641660fb7 ....?.......Af..
0090: c966f7e1668946f8837e16007538837e .f..f.F..~..u8.~
00a0: 2a007732668b461c6683c00cbb0080b9 *.w2f.F.f.......
00b0: 0100e82b00e94803a0fa7db47d8bf0ac ...+..H...}.}...
00c0: 84c074173cff7409b40ebb0700cd10eb ..t.<.t.........
00d0: eea0fb7debe5a0f97debe098cd16cd19 ...}....}.......
```

```
00e0: 6660663b46f80f824a00666a00665006 f`f;F...J.fj.fP.
00f0: 53666810000100807e02000f852000b4 Sfh.....~.... ..
0100: 41bbaa558a5640cd130f821c0081fb55 A..U.V@........U
0110: aa0f851400f6c1010f840d00fe4602b4 .............F..
0120: 428a56408bf4cd13b0f9665866586658 B.V@......fXfXfX
0130: 6658eb2a6633d2660fb74e1866f7f1fe fX.*f3.f..N.f...
0140: c28aca668bd066c1ea10f7761a86d68a ...f..f....v....
0150: 56408ae8c0e4060accb80102cd136661 V@............fa
0160: 0f8254ff81c300026640490f8571ffc3 ..T.....f@I..q..
0170: 4e544c445220202020202000000000000 NTLDR      .....
0180: 00000000000000000000000000000000 ................
0190: 00000000000000000000000000000000 ................
01a0: 000000000000000000000000d0a5265 ..............Re
01b0: 6d6f7665206469736b73206f72206f74 move disks or ot
01c0: 686572206d656469612eff0d0a446973 her media....Dis
01d0: 6b206572726f72ff0d0a507265737320 k error...Press
01e0: 616e79206b657920746f207265737461 any key to resta
01f0: 72740d0a0000000000accbd8000055aa rt...........U.
```

The sd-reader library

To read the data from the filesystem on the SD card you need the mmc_read_block primitive and an implementation of the FAT16 or FAT32 abstraction.

Because this is a week-end project, I changed gears and switched to an open source library to do the job. Roland Riegel has implemented a sd-reader library for the ATMEGA8 family. Its FAT16 implementation can fit in an ATMEGA168, but the FAT32 one requires the ROM size of an ATMEGA328.

You can download the latest source code on this page: http://www.roland-riegel.de/sd-reader/index.html.

No modifications are needed for the ATMEGA168. I just modified the configuration of the programmer to use an avrispmkii:

```
flash: $(HEX)
        avrdude -p $(MCU_AVRDUDE) -P usb -c avrispmkII -V -U flash:w:$(HEX)
```

To compile and flash:

```
make
make flash
```

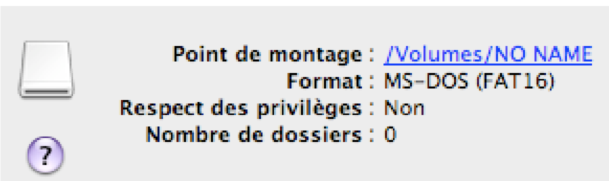
The main program of the library provides a shell on the UART at 9600 bauds with some basic commands to manipulate the SD card file system (ls, cat, cd, write).

Formatting a FAT16 filesystem on a mac

I followed the instructions provided in this Adafruit forum entry: http://forums.adafruit.com/viewtopic.php?f=31&t=7108.

```
newfs_msdos -F 16 /dev/(SD Card device/device number)
```

You can check in Disk Utility that the SD card has been correctly formatted with a FAT16 filesystem:



Testing the library

My first tries with the write command failed inexpectedly. I was able to create a directory, touch a file, but unable to write data in it...

At the same time I stumbled on the OpenLog board from Sparkfun that is based on the very same library. By comparing the implementation with the reference one, I found the explanation for my failing write. The sd-reader library expects a pin telling if the lock switch of a card is enabled. I don't provide this facility and have to bypass it.

To do so I patched the library with the modifications from the OpenLog firmware (product page) in sd_raw_config.h:

```
// From Sparkfun OpenLog
//My 2 hour pitfall: If not using card detect or write protect, assign these values:
//#define configure_pin_available() DDRC &= ~(1 << DDC4)
//#define configure_pin_locked() DDRC &= ~(1 << DDC5)
#define configure_pin_available() //Do nothing
#define configure_pin_locked() //Do nothing

//#define get_pin_available() ((PINC >> PC4) & 0x01)
//#define get_pin_locked() ((PINC >> PC5) & 0x01)
#define get_pin_available() (0) //Emulate that the card is present
#define get_pin_locked() (1) //Emulate that the card is always unlocked
```

I also added a modification of my own to display the ASCII content of a file with the cat command in main.c:

```
/* Display printable characters */
for(uint8_t i = 0; i < size; ++i)
{
  if ((buffer[i] >= 32) && (buffer[i] <= 126))
    uart_putc(buffer[i]);
  else
    uart_putc('.');
}
```

With this modification, everything is working fine:

```
$ cu -s 9600 -l /dev/tty.usbserial
manuf:  0x03
oem:    SD
prod:   SU02G
rev:    80
serial: 0x201b858a
date:   8/7
size:   1938MB
copy:   1
wr.pr.: 0/0
format: 0
free:   2030075904/2032336896
> ls
._.Trashes                          4096
.Trashes/                           0
.fseventsd/                         0
.Spotlight-V100/                    0
> mkdir test
> cd test
> ls
./                                  0
../                                 0
> touch data
> write data 0
< line 1
< line 2
< line 3
<
> ls
./                                  0
../                                 0
data                                18
> cat data
00000000: 6c 69 6e 65 20 31 6c 69 line 1li
00000008: 6e 65 20 32 6c 69 6e 65 ne 2line
00000010: 20 33 20 32 6c 69 6e 65  3
>
```

My patches to the sd-reader library are available at the end of the post.

A drawback: the required code size

After having understand how the SPI interface is working and how to dialog with an MMC card, I can see myself using the sd-reader library in a future project.

One drawback of the library is the size of code required to implement the FAT16 layer. 8kbytes is half the flash size of an ATMEGA168 without a flash loader.

layer	code size	static RAM usage
MMC/SD	2410	518
Partition	456	17
FAT16	7928	188

You also need half the size of the ATMEGA168 RAM for the MMC layer to be able to load blocks in memory.

If you need to save on flash space, an alternative is to only use raw accesses to the SD card blocks. The SD card can then be saw as a giant EEPROM where you read/write pages. The trade-off is that you will not be able to plug the card in a PC and expect to be able to read its content in the explorer/finder. It could only be read with commands issued through the MCU with a serial link for example.

For a serious application requiring FAT16 access from either MCU and PC, I would recommend to directly start a project with at least an ATMEGA328.

Source code

- mmc-spi.zip
- sd-reader_source_20090330.patch (Patch to apply on sd-reader_source_20090330.zip)

Tags: atmega168·avr·FAT

28 Comments so far ↓

- **Olivier** // Jan 25, 2010 at 12:06 pm

  What about a SD card reader ?
  http://fr.digikey.com/1/1/859090-conn-memory-sd-card-top-mnt-smd-sg5s009v1a1.html
  It's cheap, light and you can easily put the card in a computer to retrieve some data.

-  **Chris Hatton** // Mar 25, 2010 at 1:19 am

  A most excellent and enlightening article, thanks. I learned about how the SPI bus works from this. Never mind the suggestion from Olivier, some people don't seem to get the point of 'doing something for yourself'.

- **The Chief Sheep** // Apr 1, 2010 at 9:59 pm

  Hi Chris, thanks for your comment, I'm pleased to see my explanations about SPI has been helpful to someone else.
  Don't worry about Olivier. It is just a friend making fun of my posts

- **Bjorn** // Apr 3, 2010 at 5:38 pm

  Hi.
  An idea I have seen on FPGA projects that access SD-card is to format the card in PC using for example FAT16 and then generate a large , empty, file. As the generated file is the first file in the file system all data will be consecutive on the flash so by identifying where the first byte of content data is located it should be possible to start writing with the uC at that position using raw-mode. Then the file can be examinded in the PC with an intact FAT16 filesystem but with data written by the uC)
  This might be a way of storing data in FAT16 without the implementation of FAT16 file system in the uC.
  What do you think of this idea?

-  **Jens Willy Johannsen** // May 4, 2010 at 9:25 am

Hi.

Does it run ok at 16 MHz at 3.3V? Because in my datasheet it says that "Maximum frequency … is a linear curve between 2.7V and 4.5V" going from 10 MHz to 20 MHz, which results in about 13.3 MHz at 3.3V. (ATmega168-20PU, section 28.3 Speed Grades).

I'm trying to do pretty much the same as you and I would love to be able to keep my existing 16 MHz crystal.

- **The Chief Sheep** // May 5, 2010 at 11:40 pm

  Hi Jens,

  You're right, a 16MHz crystal is out of the safe operating area of the ATMEGA168.

  There are two alternatives:

  1) Use a slower crystal, 8MHz or 12MHz for example.
  2) The cleaner way is to run the ATMEGA168 at 5V and add a 3.3V low dropout regulator to separately power the SD card. You also need voltage dividers on the signal going to the card (DATA_IN, SCLK and CS). Ladyada designed its Arduino waveshield this way.

  I had neither 8 MHz or 12MHz crystal and solution 2 requires much more components so when I did my experiments I kept the 16MHz crystal and lowered the main voltage to 3.3V. I didn't encounter any problem with this setup but its not something I would suggest using for a clean design .

-  **Jens Willy Johannsen** // May 10, 2010 at 11:15 pm

  Thanks for the info, Sheep.
  I ended up using a 12 MHz crystal and everything works perfectly.

  My peripheral device (GPS) can handle 5V on VCC but serial comm runs on low-voltage TTL levels only. Which a 5V ATmega can't understand.
  And since I didn't want to clutter up the board with stuff for TTL LVTTL conversion I went out and got myself a couple of 12 MHz xtals.

- **John Ulyate** // May 29, 2010 at 5:42 am

  Your microSD pin descriptions are wrong.

  microSD Memory Card – Pins Definition

  SD Mode

  PIN NO NAME TYPE DESCRIPTION
  1 DAT2 I/O/PP DATA Line [Bit 2]
  2 CD / DAT3 I/O/PP Card Detect / DATA Line [Bit 3]
  3 CMD PP Command / Response
  4 VDD S Supply voltage
  5 CLK I Clock
  6 VSS S Supply voltage ground
  7 DAT0 I/O/PP DATA Line [Bit 0]
  8 DAT1 I/O/PP DATA Line [Bit 1]

  SPI Mode
  PIN NO NAME TYPE DESCRIPTION
  1 RSV — Reserved (*)
  2 CS I CHIP Select(Negative true)
  3 DI I DATA IN
  4 VDD S Supply voltage
  5 SCLK I Clock
  6 VSS S Supply voltage ground

7 DO O/PP DATA OUT
8 RSV — Reserved (*)

- **John Ulyate** // May 29, 2010 at 6:25 am

  Addendum:

  The SD card pinout is correct, and the uSD to SD converter will convert correctly, but from your graphics it would appear that uS D(1) is the same as SD(1) which it is not. The correct translator values is the following: (SD Bus)
  (I destroyed a translator/converter to verify)

  SD, uSD, Description
  9, 1, DAT2
  1, 2, DAT3/CD
  2, 3, CMD
  4, 4, Vdd
  5, 5, CLK
  3&6, 6, Vss
  7, 7, DAT0
  8, 8, DAT1

  BR

- **The Chief Sheep** // Jun 4, 2010 at 12:09 am

  Hi John,

  Thanks for the information. I didn't cross-checked the uSD pinout and wrongly though it was aligned with SD pinout.
  I have updated my post with a separate uSD pinout table.

- **sunil** // Sep 23, 2010 at 7:50 am

  Hi,
  Your post was very helpful for me to understand the SD SPI interface.May I know the reason why you have read the block 63 for the FAT32 MBR.I am new to the filesystem.I am also working on interfacing the SPI to SD memory card.I am able to get all the commands(CID,CSD,SCR) correctly.I tried to do a block read (the data token also has obtained) for the first 8 blocks and I could get the following data from in the first block and 0′s in all the next blocks.I know I am going wrong with my concepts .
  Can you please help me.
  00000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  00000000000000000000000000000000000000000000000000000004bc1f59e00033d
  061cfcd9f90007531e000000000000000000000000000000000000000000000000055aa

  Regards,
  Sunil

- **sunil** // Sep 23, 2010 at 8:19 am

  Hi,
  The above data from the block 0 is missing some bytes.
  The following is the data from the first block(block0)
  00000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000004bc1f59e00033d
061cfcd9f90007531e0000000000000000000000000000000000000000055aa
Can any one please help me.
Regards,

- **Kishore** // Sep 24, 2010 at 9:24 pm

  Hi Sunil,

  I am also getting the issue of yours, when ever i read a blcok i get same response as you said.

  If read to block 0, i get the all zeros and at the end few data along with the 0xaa55, if i read block 1 , one extra 0×00 adding 0xaa55 …

  Same way if i read block2 at the i am getting aa550000

  Same way if i read block3 at the end i am getting aa55000000

  i don't understand, where i am going wrong

- **Al3x 10** // Sep 3, 2011 at 10:39 pm

  i think you touched the mbr of the card…correct me if i'm wrong but the aa55 is the MBR signature.

- **Использование SD или MMC в AVR-проектах | MyLinks** // Sep 20, 2011 at 6:38 am

  [...] http://elasticsheep.com/2010/01/reading-an-sd-card-with-an-atmega168/ – это через стандартную SS-линию [...]

- **Nick Bansal** // May 1, 2012 at 7:13 pm

  Sorry I am not understanding all the components for this… Am i just being blind or is there components on the breadboard that aren't on the schematic?…. im quite new to this so can i have some help on the matter please

- **The Chief Sheep** // Jun 4, 2012 at 7:49 pm

  @Nick,
  The schematic illustrates the connection of the SD card to the SPI pins of the ATMEGA168.
  On the breadboard, the green PCB is a breakout board that provides an ISP connector to ease the programming of the ATMEGA168 (Older version of http://www.sparkfun.com/products/8508). The red led was only added to help me debug my programs.
  BR
  Mathieu

- **Maryam** // Jun 16, 2012 at 3:56 pm

  Hi,
  thank you very much for your really helpful post. I have a project about SD card and I want to use your idea of powering the micro controller with 3.3V can I use this idea for ATMEGA16A?

- **The Chief Sheep** // Jun 18, 2012 at 9:21 pm

  Hi Maryam,
  It should work fine.
  You would only be limited by the maximum safe operating frequency of the ATMEGA16A. It is about 10.67MHz at 3.3V (see datasheet 27.3).
  BR
  Mathieu

- **muhereza shafiki** // Jun 28, 2012 at 11:07 am how can i build an mp3 micro sd player out of only transistors not ics

- **The Chief Sheep** // Jun 28, 2012 at 10:23 pm

  Hi Muhereza,
  It should be theoretically possible but you would need a really big bag of transistor!
  BR
  Mathieu

- **harry** // Jul 12, 2012 at 7:28 am

  How to Atmega 128 microcontroller read wav file using the code vision avr ?

- **The Chief Sheep** // Jul 13, 2012 at 9:12 pm

  Hi Harry,
  I don't know much about the CodeVision IDE as I am using avr-gcc and avr-libc.
  I think you should be able to import the source code of my project into it without too much difficulty.
  But you may encounter syntax problems with the compiler.
  BR
  Mathieu

- **Edson** // Nov 26, 2012 at 3:09 pm

  I would like to know if is it possible to build this SD card reader using a common PIC like 18F series, because the PIC here used is difficult to find.
  Please, give me some advice.
  Thanks.

- **Badr A.** // Apr 18, 2013 at 5:21 pm

  A lot of thanks for the helpful posted. I'm really interested in what have posted. I have a project that need from me to write to a text file in the SD card using the same microcontoller ATMega168. I would like to know what I should change in the source you have posted.

- **The Chief Sheep** // Apr 26, 2013 at 7:10 pm

  Hi Badr,
  The sd-reader library has functions to write files on a FAT filesystem.
  To write a text file, you can use the fat_create_file, fat_write_file and fat_close_file functions.
  Manipulating the fat_dir_struct and fat_file_struct is not very intuitive but you can follow the example code in sd-reader main.c for the "write" command.
  BR
  Mathieu

- **Matthew** // Aug 31, 2013 at 10:15 am

  Hi

  I've just bought a FlashAir wifi enabled SD card and was wondering if there is a way to power the SD card directly with a 3.3V battery just to enable the wifi so that I can transfer pictures to and from it without having to connect it to a camera.

- **The Chief Sheep** // Oct 12, 2013 at 11:08 pm

  Hi Matt,
  I think it should be possible.
  You can use the VDD and VSS1/VSS2 pins to respectively connect a 3.3V regulated power supply and ground to your SD card.
  Best regards,
  Mathieu

Leave a Comment

Name

Mail

Website

Submit

- 

- ## Search It!

To search, type and hit e

- ## Recent Entries

  - Bluetooth module firmware dump6.28
  - Bluetooth module parallel-SPI interface6.28
  - New toy6.18
  - Babyphone hardware design1.25
  - Babyphone audio playback1.21
  - Current project9.27
  - Bluetooth module breakout boards are back in stock9.26
  - Serial Bluetooth module – Master/Slave connection5.20
  - Serial Bluetooth module slave test5.8
  - Bluetooth module breakout boards received5.2
  - Visit the archives for more!