Samsung SD & MicroSD Card product family

SDA 3.0 specification compliant-Up to High Speed mode

datasheet

SAMSUNG ELECTRONICS RESERVES THE RIGHT TO CHANGE PRODUCTS, INFORMATION AND SPECIFICATIONS WITHOUT NOTICE.

Products and specifications discussed herein are for reference purposes only. All information discussed herein is provided on an "AS IS" basis, without warranties of any kind.

This document and all information discussed herein remain the sole and exclusive property of Samsung Electronics. No license of any patent, copyright, mask work, trademark or any other intellectual property right is granted by one party to the other party under this document, by implication, estoppel or otherwise.

Samsung products are not intended for use in life support, critical care, medical, safety equipment, or similar applications where product failure could result in loss of life or personal or physical harm, or any military or defense application, or any governmental procurement to which special terms or provisions may apply.

For updates or additional information about Samsung products, contact your nearest Samsung office.

All brand names, trademarks and registered trademarks belong to their respective owners.

Revision History

Revision No.	History	Draft Date	Remark	Editor
1.0	1. Customer sample acquired	Jun, 02. 2011	Final	S.M.Lee

Table Of Contents

1.0 INFORMATION

M	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**-**	**X**	**X**	**X**	**X**	**X**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**

1. Module: M

2. Module Configuration
C : Flash Card (SLC)
E : Flash Card (OneNAND)
M : Flash Card (MLC)

3~4. Flash Density
64 : 64M 28 : 128M
56 : 256M 2 : 512M
5D : 512M DDP 1G : 1G
1D : 1G DDP 2G : 2G
2D : 2G DDP 4G : 4G
8G : 8G AG : 16G
B3 : 32Gb 3bit BG: 32Gb 2bit
BA : 32Gb DDR BT : 32G(3Bit Toggle)

5. Feature
R : microSD
F : SD

6~8. Card Density
008 : 8M Byte 016 : 16M Byte
032 : 32M Byte 048 : 48M Byte
064 : 64M Byte 096 : 96M Byte
128 : 128M Byte 192 : 192M Byte
256 : 256M Byte 384 : 384M Byte
512 : 512M Byte 01G : 1G Byte
02G : 2G Byte 04G : 4G Byte
08G : 8G Byte 16G : 16G Byte
32G : 32G Byte

9. Card Type
U : microSD
W : SD

10. Component Generation
M : 1st Generation A : 2nd Generation
B : 3rd Generation C : 4th Generation
D : 5th Generation E : 6th Generation

11. Flash Package
C : CHIP Y : TSOP1
V : WSOP B : TBGA

12. PCB Revision and Production site.
A : None (SEC) B : 1st Rev. (SEC)
C : 2nd Rev. (SEC) D : 3rd Rev. (SEC)
P : None (STS) Q : 1st Rev. (STS)
R : 2nd Rev. (STS) U : None(ATP)
V : 1st Rev.(ATP) W : 2nd Rev.(ATP)
Y : None(SPIL)

13. " - "

14. Packing Type
0 : With Label
1 : With Label/Contents
2 : No Label
3 : With Label Class2
4 : With Label Class4
5 : With Label Class6
6 : With Label Class10
7 : With Label UHS-I Speed Class1
A : None
B : Blue
D : Dark Black
G : Gray
H : White
M : Module Type
N : Navy Blue
P : Class2(No Label)
Q : Class4(No Label)
R : Class6(No Label)
S : Class10(No Label)
T : Metal Blue
W : Wine

15 ~ 16. Controller
ME: SS6651ACWWE

17 ~ 18. Customer Grade
" Customer List Reference "

2.0 PRODUCT LINE-UP

Model Number	Capacities	Remarks
MMBTF04GWBCA-xMExx	4GB	SD Card (x : Refer to the Ordering Information)
MMBTF08GWBCA-xMExx	8GB	
MMBTF16GWBCA-xMExx	16GB	

3.0 INTRODUCTION

3.1 General Description

The SD/microSD is a memory card that is specifically designed to meet the security, capacity, performance and enviroment requirements inherent in newly emerging audio and video consumer electronic devices. The SD/microSD will include a copyright protection mechanism that complies with the security of the SDMI standard and will be faster and capable for higher Memory capacity. The SD/microSD security system uses mutual authentication and a "new ciper algorithm" to protect from illegal usage of the card content. A none secured access to the user’s own content is also available.
The SD/ microSD communication is based on an advanced 9 and 8-pin interface (SD:9pin, microSD:8pin)) designed to operate in at maximum operating frequency of 208MHz and 2.7V ~ 3.6V voltage range with 2 Type signaling(1.8V & 3.3V)*. More detail informations on the interface, and mechanical description is defined as a part of this specification.

* High Speed mode Limited on this Specification.

3.2 System Features

- Compliant with SD Memory Card Specifications PHYSICAL LAYER SPECIFICATION Version 3.00
  - Based on SD Memory Card Specification 3.0 compatible Test Device.
  - Bus speed only support up to High Speed Mode (3.3V signaling, frequency up to 50MHz)

- Targeted for portable and stationary applications
- Memory capacity:
  1) Standard Capacity SD Memory Card(SDSC) : Up to and including 2 GB
  2) High Capacity SD Memory Card(SDHC) : More than 2GB and up to and including 32GB
  3) Extended Capacity SD Memory Card(SDXC) : More than 32GB and up to and including 2TB

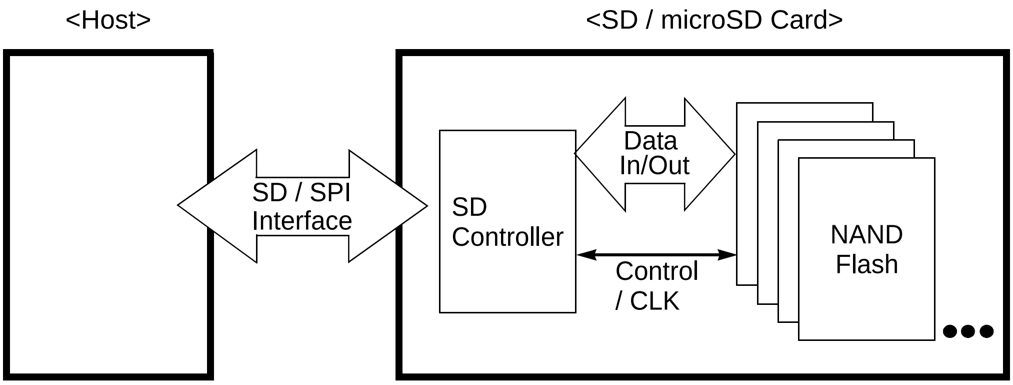
- Voltage range:
  High Voltage SD Memory Card – Operating voltage range: 2.7-3.6 V

- Designed for read-only and read/write cards.

- Bus Speed Mode
  1) Default mode: Variable clock rate 0 - 25 MHz, up to 12.5 MB/sec interface speed (using 4 parallel data lines)
  2) High-Speed mode: Variable clock rate 0 - 50 MHz, up to 25 MB/sec interface speed (using 4 parallel data lines)

- Switch function command supports High-Speed, and future functions
- Correction of memory field errors
- Card removal during read operation will never harm the content
- Content Protection Mechanism - Complies with highest security of SDMI standard.
- Password Protection of cards (CMD42 - LOCK_UNLOCK)
- Write Protect feature using mechanical switch
- Built-in write protection features (permanent and temporary)
- Card Detection (Insertion/Removal)
- Application specific commands
- Comfortable erase mechanism
- Weight : SD Card Max. 2.5g / microSD Card Max. 1g

3.3 System Block Diagram



4.0 PRODUCT SPECIFICATION

4.1 Current Consumption

This information table below provides current consumption of Samsung SD/microSD Card. Current consumption is measured by averaging over 1 second.

[Table 4-1] : Current Consumption Table

Mode	Max. Interface Frequency	Operations	Max.
Default Mode	25Mhz	Read	100mA
		Write	
High Speed Mode	50Mhz	Read	200mA
		Write	

NOTE:
Current consumption on each device can be varied by NAND Flash, . of chips, test conditions and Etc. For specific information, refer to Samsung SD/microSD Card Qualification report.

4.2 System Performance

4.2.1 Product Performance & Speed Class Information

Product Performance and Speed Class Informations are based on TestMetrix compliance Tool. Note that the performance measured by TestMetrix does not represent real performance in various circumstances.

[Table 4-2] : Performance Information

Product Number	Write Performance (MB/s)	Read Performance (MB/s)	Speed Class[1]
MMBTF04GWBCA-xMExx	7	24	Class 4
MMBTF08GWBCA-xMExx	13		Class 6
MMBTF16GWBCA-xMExx	13		Class 6

NOTE:
1) Five Speed Classes are defined and indicate minimum performance of the cards in Speed Class Test Mode. Speed Class compliant SDA Physical Layer Specification, Version 3.00

.Class 0 - These Class cards do not specify performance. It includes all the legacy cards prior to this specification, regardless of its performance
.Class 2 - is more than or equal to 2MB/s performance
.Class 4 - is more than or equal to 4MB/s performance
.Class 6 - is more than or equal to 6MB/s performance
.Class 10 - is more than or equal to 10MB/s performance

4.2.2 Read, Write Timeout Error Conditions

SEC SD/microSD Card shall complete the command within the time period defined as follows or give up and return and error message. If the host does not get any response with the given timeout it should assume that the card is not going to respond and try recover. For more information, refer to Section 4.6 of the SDA Physical Layer Specification, Version 3.00

[Table 4-3] : Timeout Error Conditions

Timing	Max. Value
Block Read Access Time	100ms
Block Write Access Time	250ms(SDSC/SDHC), 500ms(SDXC)
Initialization Time out(ACMD 41)[1]	1s

NOTE:
1) The host shall set ACMD41 timeout more than 1 second to abort repeat of issuing ACMD41 when the card does not indicate ready. The timeout count starts from the first ACMD41 which is set voltage window in the argument.

4.3 SD Mode Card Registers

Six registers are defined within the card interface: OCR, CID, CSD, RCA, DSR and SCR. These can be accessed only by corresponding commands. The OCR, CID, CSD and SCR registers carry the card/content specific information, while the RCA and DSR registers are configuration registers storing actual configuration parameters.

4.3.1 OCR Register

• The 32-bit operation conditions register stores the VDD voltage profile of the card. Additionally, this register includes status information bits.
•See Section 5.1 of the SDA Physical Layer Specification, Version 3.00 for more information.

[Table 4-4] : OCR Register Definition

OCR bit	VDD Voltage Window	OCR Value
0-3	reserved	0
4	reserved	0
5	reserved	0
6	reserved	0
7	reserved for Low Voltage Range	0
8	reserved	0
9	reserved	0
10	reserved	0
11	reserved	0
12	reserved	0
13	reserved	0
14	reserved	0
15	2.7 - 2.8	1
16	2.8 - 2.9	1
17	2.9 - 3.0	1
18	3.0 - 3.1	1
19	3.1 - 3.2	1
20	3.2 - 3.3	1
21	3.3 - 3.4	1
22	3.4 - 3.5	1
23	3.5 - 3.6	1
24[3]	Switching to 1.8V Accepted (S18A)	0
24 - 29	reserved	0
30	Card Capacity Staus(CCS)[1]	-
31	Card power up status bit(busy)[2]	-

NOTE:
1) This bit is valid only when the card power up status bit is set.
2) This bit is set to LOW if the card has not finished the power up routine.
3) Only UHS-I card supports this bit.

4.3.2 CID Register

The Card IDentification (CID) register is 128 bits wide. It contains the card identification information used during the card identification phase. Every individual Read/Write (RW) card shall have an unique identification number. It is programmed during manufacturing and cannot be changed by card hosts. The structure of the CID register is defined in the following paragraphs:

[Table 4-5] : CID Register Fields

Name	Field	Type	Width	CID Value		
				4GB	8GB	16GB
Manufacturer ID	MID	Binary	8	CID Register Value can be provided by Customer Request		
OEM/Application ID	OID	ASCII	16			
Product name	PNM	ASCII	40			
Product revision	PRV	BCD	8			
Product serial number	PSN	Binary	32			
Reserved	-	-	4			
Manufacturing date	MDT	BCD	12			
CRC7 checksum	CRC	Binary	7			
not used, always ’1’	-	-	1			

4.3.3 CSD Register (CSD Version 1.0)

The Card-Specific Data register provides information on how to access the card contents. The CSD defines the data format, error correction type, maximum data access time, data transfer speed, whether the DSR register can be used etc. The programmable part of the register (entries marked by W or E, see below) can be changed by CMD27. The type of the entries in the table below is coded as follows: R = readable, W(1) = writable once, W=multiple writable.

[Table 4-6] : The CSD Register Fields (CSD Version 1.0)

Name	Field	Width	Cell Type	CSD-slice	CSD Value		
					4GB	8GB	16GB
CSD structure	CSD_STRUCTURE	2	R	[127:126]	Version 1.0		
Reserved	-	6	R	[125:120]	-		
Data read access-time 1	TAAC	8	R	[119:112]	N/A		
Data read access-time 2 in CLK cycles (NSAC*100)	NSAC	8	R	[111:104]	N/A		
Max. Data transfer rate	TRAN_SPEED	8	R	[103:96]	N/A		
Card command classes	CCC	12	R	[95:84]	N/A		
Max. read data block length	READ_BL_LEN	4	R	[83:80]	N/A		
Partial blocks for read allowed	READ_BL_PARTIAL	1	R	[79:79]	N/A		
Write block misalignment	WRITE_BLK_MISALIGN	1	R	[78:78]	N/A		
Read block misalignment	READ_BLK_MISALIGN	1	R	[77:77]	N/A		
DSR implemented	DSR_IMP	1	R	[76:76]	N/A		
Reserved	-	2	R	[75:74]	-		
Device size	C_SIZE	12	R	[73:62]	N/A		
Max. read current @ V_{DD} min	VDD_R_CURR_MIN	3	R	[61:59]	N/A		
Max. read current @ V_{DD} max	VDD_R_CURR_MAX	3	R	[58:56]	N/A		
Max. write current @ V_{DD} min	VDD_W_CURR_MIN	3	R	[55:53]	N/A		
Max. write current @ V_{DD} max	VDD_W_CURR_MAX	3	R	[52:50]	N/A		
Device size multiplier	C_SIZE_MULT	3	R	[49:47]	N/A		
Erase single block enable	ERASE_BLK_EN	1	R	[46:46]	N/A		
Erase sector size	SECTOR_SIZE	7	R	[45:39]	N/A		
Write protect group size	WP_GRP_SIZE	7	R	[38:32]	N/A		
Write protect group enable	WP_GRP_ENABLE	1	R	[31:31]	N/A		
Reserved (Do Not Use)		2	R	[30:29]	-		
Write speed factor	R2W_FACTOR	3	R	[28:26]	N/A		
Max. write data block length	WRITE_BL_LEN	4	R	[25:22]	N/A		
Partial blocks for write allowed	WRITE_BL_PARTIAL	1	R	[21:21]	N/A		
Reserved	-	5	R	[20:16]	-		
File format group	FILE_FORMAT_GRP	1	R/W(1)	[15:15]	N/A		
Copy flag (OTP)	COPY	1	R/W(1)	[14:14]	N/A		
Permanent write protection	PERM_WRITE_PROTECT	1	R/W(1)	[13:13]	N/A		
Temporary write protection	TMP_WRITE_PROTECT	1	R/W	[12:12]	N/A		
File format	FILE_FORMAT	2	R/W(1)	[11:10]	N/A		
Reserved		2	R/W	[9:8]	-		
CRC	CRC	7	R/W	[7:1]	N/A		
Not used, always ’1’	-	1	-	[0:0]	-		

4.3.4 CSD Register (CSD Version 2.0)

The following Table shows Definition of the CSD Version 2.0 for the High Capacity SD Memory Card and Extended Capacity SD Memory Card. The following sections describe the CSD fields and the relevant data types for the High Capacity SD Memory Card.

CSD Version 2.0 is applied to only the High Capacity SD Memory Card. The field name in parenthesis is set to fixed value and indicates that the host is not necessary to refer these fields. The fixed values enables host, which refers to these fields, to keep compatibility to CSD Version 1.0. The Cell Type field is coded as follows: R = readable, W(1) = writable once, W = multiple writable.

[Table 4-7] : The CSD Register Fields (CSD Version 2.0)

Name	Field	Width	Cell Type	CSD-slice	CSD Value		
					4GB	8GB	16GB
CSD structure	CSD_STRUCTURE	2	R	[127:126]	CSD Version 2.0		
Reserved	-	6	R	[125:120]	-		
Data read access-time	(TAAC)	8	R	[119:112]	1000.00 us		
Data read access-time in CLK cycles (NSAC*100)	(NSAC)	8	R	[111:104]	0 cycles		
Max. Data transfer rate	(TRAN_SPEED)	8	R	[103:96]	25 Mbit/s or 50 Mbit/s		
Card command classes	CCC	12	R	[95:84]	class 0 2 4 5 7 8 10		
Max. read data block length	(READ_BL_LEN)	4	R	[83:80]	512 bytes		
Partial blocks for read allowed	(READ_BL_PARTIAL)	1	R	[79:79]	0		
Write block misalignment	(WRITE_BLK_MISALIGN)	1	R	[78:78]	0		
Read block misalignment	(READ_BLK_MISALIGN)	1	R	[77:77]	0		
DSR implemented	DSR_IMP	1	R	[76:76]	0		
Reserved	-	6	R	[75:70]	-		
Device size	C_SIZE	22	R	[69:48]	7579	15191	30531
Reserved	-	1	R	[47:47]	-		
Erase single block enable	(ERASE_BLK_EN)	1	R	[46:46]	1		
Erase sector size	(SECTOR_SIZE)	7	R	[45:39]	128 blocks		
Write protect group size	(WP_GRP_SIZE)	7	R	[38:32]	1 sectors		
Write protect group enable	(WP_GRP_ENABLE)	1	R	[31:31]	0		
Reserved	-	2	R	[30:29]	-		
Write speed factor	(R2W_FACTOR)	3	R	[28:26]	4		
Max. write data block length	(WRITE_BL_LEN)	4	R	[25:22]	512 bytes		
Partial blocks for write allowed	(WRITE_BL_PARTIAL)	1	R	[21:21]	0		
Reserved	-	5	R	[20:16]	-		
File format group	(FILE_FORMAT_GRP)	1	R	[15:15]	0		
Copy flag (OTP)	COPY	1	R/W(1)	[14:14]	0		
Permanent write protection	PERM_WRITE_PROTECT	1	R/W(1)	[13:13]	0		
Temporary write protection	TMP_WRITE_PROTECT	1	R/W	[12:12]	0		
File format	(FILE_FORMAT)	2	R	[11:10]	0		
Reserved		2	R	[9:8]	-		
CRC	CRC	7	R/W	[7:1]	-		
Not used, always '1'	-	1	-	[0:0]	-		

4.3.5 RCA Register

The writable 16-bit relative card address register carries the card address that is published by the card during the card identification. This address is used for the addressed host-card communication after the card identification procedure. The default value of the RCA register is 0x0000. The value 0x0000 is reserved to set all cards into the *Stand-by State* with CMD7.

4.3.6 SCR Register

In addition to the CSD register, there is another configuration register named SD CARD Configuration Register (SCR). SCR provides information on the SD Card’s special features that were configured into the given card. The size of SCR register is 64bits. The register shall be set in the factory by the SD Card manufacturer. The following table describes the SCR register content.

[Table 4-8] : The SCR Fields

Name	Field	Width	Cell Type	SCR-slice	SCR Value		
					4GB	8GB	16GB
SCR structure	SCR_STRUCTURE	4	R	[63:60]	SCR Version 1.0		
SD Memory Card - Spec. Version	SD_SPEC	4	R	[59:56]	Version 2.00 or Version 3.00		
data_status_after erases	DATA_STAT_AFTER_ERASE	1	R	[55:55]	0		
SD Security Support	SD_SECURITY	3	R	[54:52]	Version 2.00		
DAT Bus widths supported	SD_BUS_WIDTHS	4	R	[51:48]	1 bit(DAT0) + 4 bit(DAT0-3)		
Spec. Version 3.00 or Higher	SD_SPEC3	1	R	[47]	Version 3.00		
Reserved		13	R	[46:34]	-		
Command Support bits	CMD_SUPPORT	14	R	[33:32]	Not Supported		
Reserved for manufacturer usage	-	32	R	[31:0]	-		

4.3.7 SD Status Register

The SD Status contains status bits that are related to the SD Memory Card proprietary features and may be used for future application specific usage. The size of the SD Status is one data block of 512bit. The content of this register is transmitted to the Host over the DAT bus along with 16bit CRC. The SD Status is sent to the host over the DAT bus if ACMD13 is sent (CMD55 followed with CMD13). ACMD13 can be sent to a card only in 'tran_state' (card is selected). SD Status structure is described in below. Unused reserved bits shall be set to 0.

[Table 4-9] : SD Status Register

Bits	Field	Cell Type	Data			Value		
			4GB	8GB	16GB	4GB	8GB	16GB
511:510	DATA_BUS_WIDTH	S R	0x00			1bit width or 4bit width		
509	SECURED_MODE	S R	0x00			-		
508:502	Reserved for Security Functions (Refer to Part 3 Security Specification)							
501:496	Reserved							
495:480	SD_CARD_TYPE	S R	0x0000			Regular SD RD/WR Card		
479:448	SIZE_OF_PROTECTED_AREA	S R	0x2000000	0x3000000	0x4000000	-		
447:440	SPEED_CLASS	S R	0x2	0x3		Class 4	Class 6	
439:432	PERFORMANCE_MOVE	S R	0x2	0x3		2 [MB/sec]	3 [MB/sec]	
431:428	AU_SIZE	S R	0x9			4MB		
427:424	Reserved							
423:408	ERASE_SIZE	S R	0x10			16 AU		
407:402	ERASE_TIMEOUT	S R	0x1			1sec		
401:400	ERASE_OFFSET	S R	0x2			2sec		
399:312	Reserved							
311:0	Reserved for Manufacturer							

NOTE:
Speed Class that supports Class 10 shall not use the Pm value stored in the SD Status to calculate performance in any fragmented AU. Class 10 Performance is defined only for entirely free AUs

4.4 SPI Mode Card Registers

Unlike the SD Memory card protocol (where the register contents is sent as command response),reading the contents of the CSD and CID registers in SPI mode is a simple read-block transaction. The card will respond with a standard response token followed by data block of 16 bytes suffixed with a 16-bit CRC.
The data timeout for the CSD command cannot be set to the cards TAAC since this value is stored in the card's CSD. Therefore, the standard response timeout value(NCR) is used for read latency of the CSD register.

4.5 User Capacity

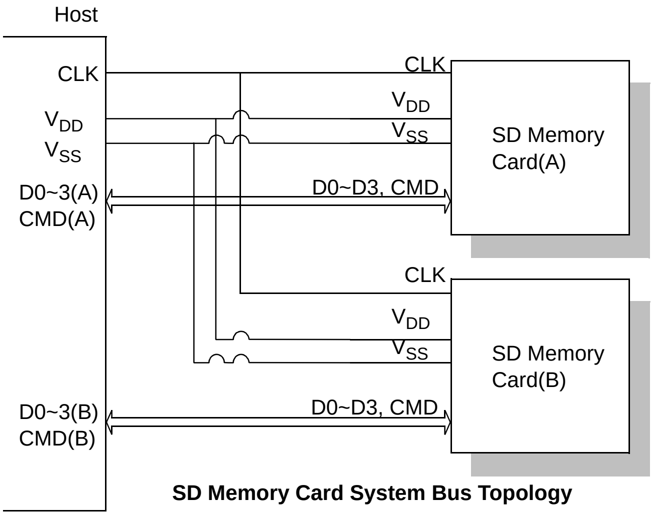
This information table below provides user capacity of Samsung SD/microSD Card.
Product user density is based on SD Formatter 2.0 tool with FAT File system.
SD Formatter 2.0 software formats all SD Cards and SDHC Cards using a formatting program that complies with official SD memory card requirements.

Product Number	File System	Tot. Sector No.	User Capacity[Byte]
MMBTF04GWBCA-xMExx	FAT32	7,745,536	3,965,714,432
MMBTF08GWBCA-xMExx		15,540,224	7,956,594,688
MMBTF16GWBCA-xMExx		31,248,384	15,999,172,608

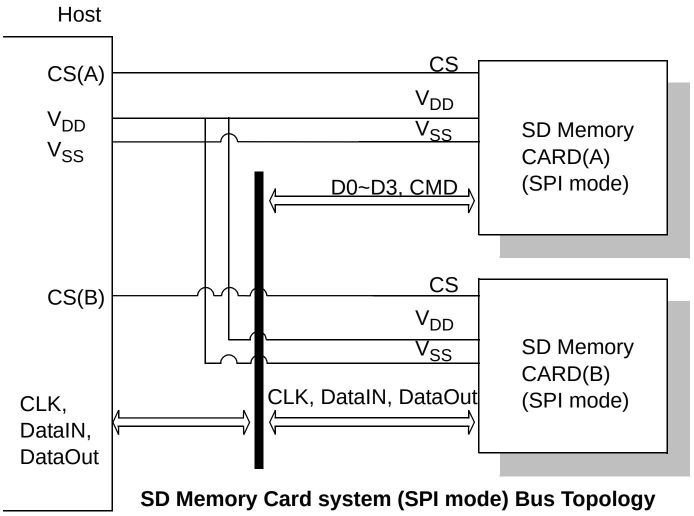
NOTE :
SD or SDHC Card file systems formatted with generic operating system formatting software do not comply with official SD memory card requirement and optimum performance may not be experienced

5.0 INTERFACE DESCRIPTION

5.1 SD/microSD SD mode Bus Topology / SD/microSD SPI Bus Topology



SD Memory Card System Bus Topology



SD Memory Card system (SPI mode) Bus Topology

The SD/microSD Memory Card system defines two alternative communication protocols: SD and SPI. The host system can choose either one of modes. The card detects which mode is requested by the host when the reset command is received and expects all further communication to be in the same communication mode. Common bus signals for multiple card slots are not recommended. A single SD bus should connect a single SD card. Where the host system supports a high-speed mode, a single SD bus shall be connected to a single SD card.

The SD/microSD bus includes the following signals:

- CMD : Bidirectional Command/Response signal
- DAT0 - DAT3 : 4 Bidirectional data signals
- CLK : Host to card clock signal
- V_{DD}, $V_{SS1,}$ V_{SS2}: Power and ground signals

The SD/microSD Card bus has a single master (application), multiple slaves (cards), synchronous start topology (refer to Figure 5-2). Clock, power and ground signals are common to all cards. Command (CMD) and data (DAT0-DAT3) signals are dedicated to each card providing continues point to point connection to all the cards.

During initialization process, commands are sent to each card individually, allowing the application to detect the cards and assign logical addresses to the physical slots. Data is always sent (received) to (from) each card individually. However, in order to simplify the handling of the card stack, after initialization process, all commands may be sent concurrently to all cards. Addressing information is provided in the command packet.

SD Bus allows dynamic configuration of the number of data lines. After power-up, be default, the SD/microSD Card will use only DAT0 for data transfer. After initialization, the host can change the bus width(number of active data lines). This feature allows and easy trade off between hardware cost and system performance. Note that while DAT1-DAT3 are not in use, the related Host's DAT lines should be in tri-state (input mode). For SDIO cards DAT1 and DAT2 are used for signaling.

The SPI compatible communication mode of the SD/microSD Memory Card is designed to communicate with a SPI channel, commonly found in various microcontrollers in the market. The interface is selected during the first reset command after power up and cannot be changed as long as the part is powered on.
The SPI standard defines the physical link only, and not complete data transfer protocol. The SD/microSD Card SPI implementation uses the same command set of the SD mode. From the application point of view, the advantage of the SPI mode is the capability of using an off-the-shelf host, hence reducing the design-in effort to minimum. The disadvantage is the loss of performance, relatively to the SD mode which enables the wide bus option.

The SD/microSD Card SPI interface is compatible with SPI hosts available on the market. As any other SPI device the SD/microSD Card SPI channel consists the following four signals:

- CS : Host to card Chip Select signal
- CLK : Host to card clock signal
- DataIN : Host to card data signal
- DataOut: Card to host data signal

Another SPI common characteristic is byte transfers, which is implemented in the card as well. All data tokens are multiples of bytes (8 bit) and always byte aligned to the CS signal.

The card identification and addressing methods are replaced by a hardware Chip Select (CS) signal. There are no broadcast commands. For every command, a card (slave) is selected by asserting (active low) the CS signal .

The CS signal must be continuously active for the duration of the SPI transaction (command, response and data). The only exception occurs during card programming, when the host can de-assert the CS signal without affecting the programming process.

5.2 Bus Protocol

5.2.1 SD Bus

For more details, refer to Section 3.6.1 of the SDA Physical Layer Specification, Version 3.00

5.2.2 SPI Bus

For more details, refer to Chapter 7 of the SDA Physical Layer Specification, Version 3.00

5.3 SD/microSD Card Pin Assignment

5.3.1 SD Card Pin Assignment

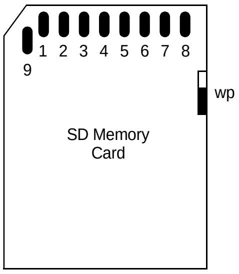


Figure 5-1. SD Memory Card shape and interface (top view)

The SD Memory Card has the form factor 24 mm x 32 mm x 2.1 mm or 24 mm x 32 mm x 1.4 mm.
Figure5-1 shows the general shape of the shape and interface contacts of the SD Memory Card. The detailed physical dimensions and mechanical description are given in section 5.4.

The following Table defines the card contacts:

[Table 5-1] : SD Memory Card Pad Assignment

Pin #	Name	Type[1]	Description	Name	Type	Description
SD Mode				SPI Mode		
1	CD/DAT3[2]	I/O/PP[3]	Card Detect / Data Line [Bit 3]	CS	I[3]	Chip Select (neg true)
2	CMD	PP	Command/Response	DI	I	Data In
3	VSS1	S	Supply voltage ground	VSS	S	Supply voltage ground
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	VSS2	S	Supply voltage ground	VSS2	S	Supply voltage ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO	O/PP	Data Out
8	DAT1[4]	I/O/PP	Data Line [Bit 1]	RSV		
9	DAT2[5]	I/O/PP	Data Line [Bit 2]	RSV		

NOTE:
1) S: power supply; I: input; O: output using push-pull drivers; PP: I/O using push-pull drivers;
2) The extended DAT lines (DAT1-DAT3) are input on power up. They start to operate as DAT lines after SET_BUS_WIDTH command. The Host shall keep its own DAT1-DAT3 lines in input mode, as well, while they are not used.
3) At power up this line has a 50KOhm pull up enabled in the card. This resistor serves two functions Card detection and Mode Selection. For Mode Selection, the host can drive the line high or let it be pulled high to select SD mode. If the host wants to select SPI mode it should drive the line low. For Card detection, the host detects that the line is pulled high. This pull-up should be disconnected by the user, during regular data transfer, with SET_CLR_CARD_DETECT (ACMD42) command
4) DAT1 line may be used as Interrupt Output (from the Card) in SDIO mode during all the times that it is not in use for data transfer operations (refer to "SDIO Card Specification" for further details).
5) DAT2 line may be used as Read Wait signal in SDIO mode (refer to "SDIO Card Specification" for further details).

5.3.2 microSD Card Assignment

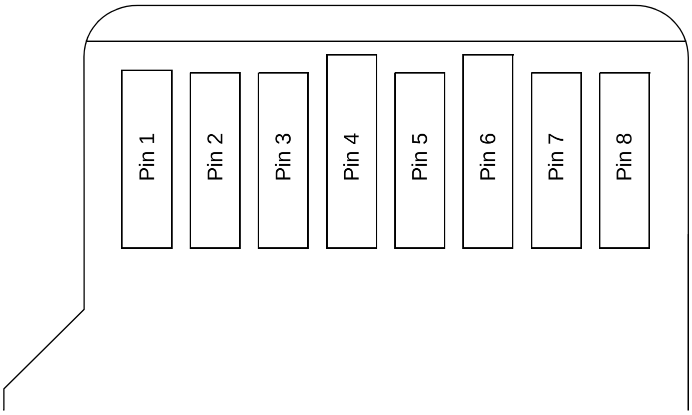


Figure 5-2. Contact Area

[Table 5-2] : microSD Contact Pad Assignment

Pin #	SD Mode			SPI Mode		
	Name	Type[1]	Description	Name	Type[1]	Description
1	DAT2[2.5]	I/O/PP	Data Line [Bit 2]	RSV		Reserved
2	CD/DAT3[2]	I/O/PP[3]	Card Detect / Data Line [Bit 3]	CS	I[3]	Chip Select (neg true)
3	CMD	PP	Command/Response	DI	I	Data In
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	VSS	S	Supply voltage ground	VSS	S	Supply voltage ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO	O/PP	Data Out
8	DAT1[2.4]	I/O/PP	Data Line [Bit 1]	RSV[4]		

NOTE:
1) S: power supply; I: input; O: output using push-pull drivers; PP: I/O using push-pull drivers ;
2) The extended DAT lines (DAT1-DAT3) are input on power up. They start to operate as DAT lines after SET_BUS_WIDTH command. The Host shall keep its own DAT1-DAT3 lines in input mode, as well, while they are not used.
3) At power up this line has a 50KOhm pull up enabled in the card. This resistor serves two functions Card detection and Mode Selection. For Mode Selection, the host can drive the line high or let it be pulled high to select SD mode. If the host wants to select SPI mode it should drive the line low. For Card detection, the host detects that the line is pulled high. This pull-up should be disconnected by the user, during regular data transfer, with SET_CLR_CARD_DETECT (ACMD42)
4) DAT1 line may be used as Interrupt Output (from the Card) in SDIO mode during all the times that it is not in use for data transfer operations (refer to "SDIO Card Specification" for further details).
5) DAT2 line may be used as Read Wait signal in SDIO mode (refer to "SDIO Card Specification" for further details).

5.4 Mechanical Specification

This section describes the mechanical and electrical features, as well as SEC SD/microSD Card environmental reliability and durability specifications. For more details you can refer to Chapter 8 of SDA Physical Layer Specification, Version 2.00 and SDA ,microSD Card Addendum, Section 3.0 Mechanical Specification for microSD Memory Card.

5.4.1 Mechanical Form Factor of microSD

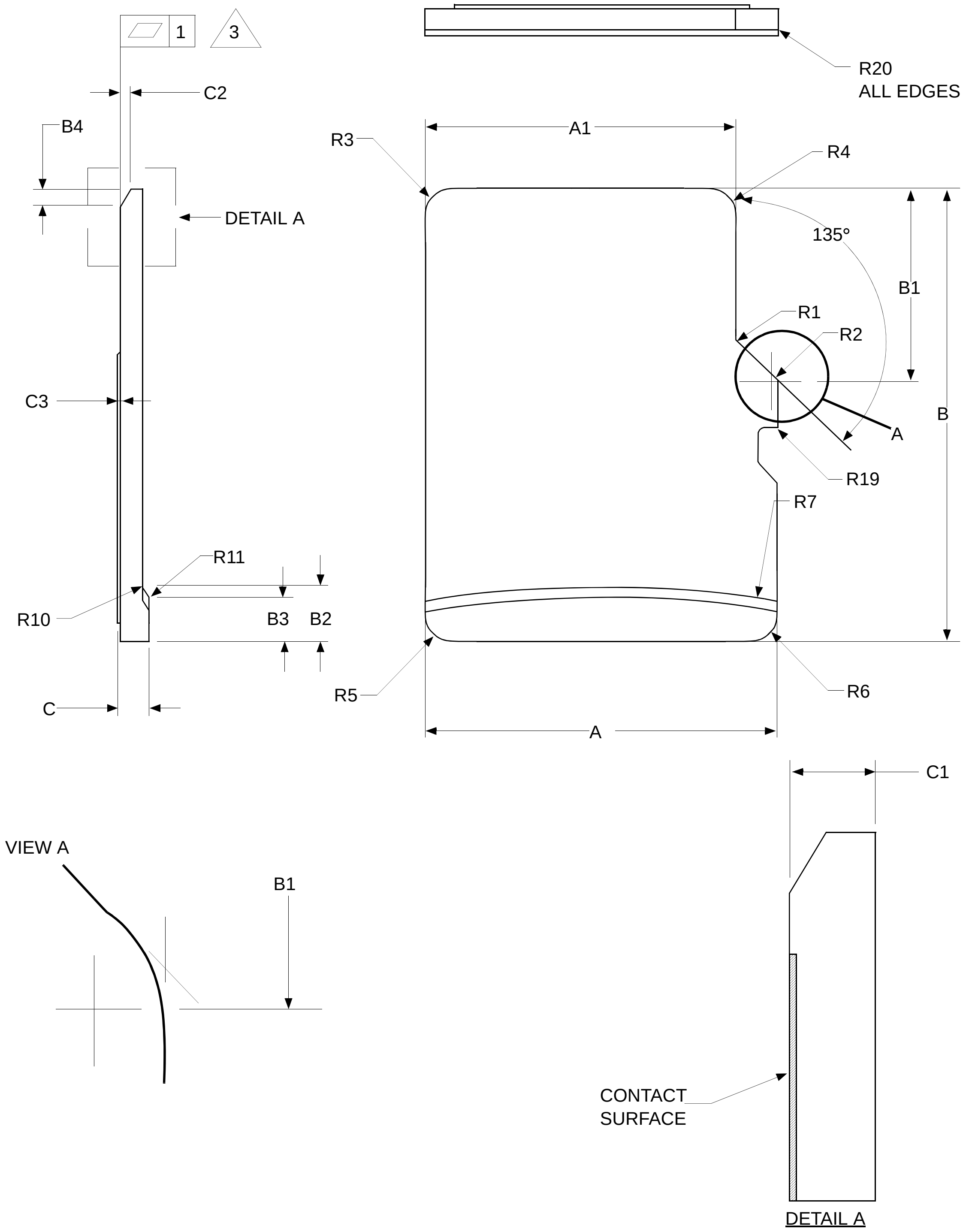


Figure 5-3. Mechanical Description: Top View

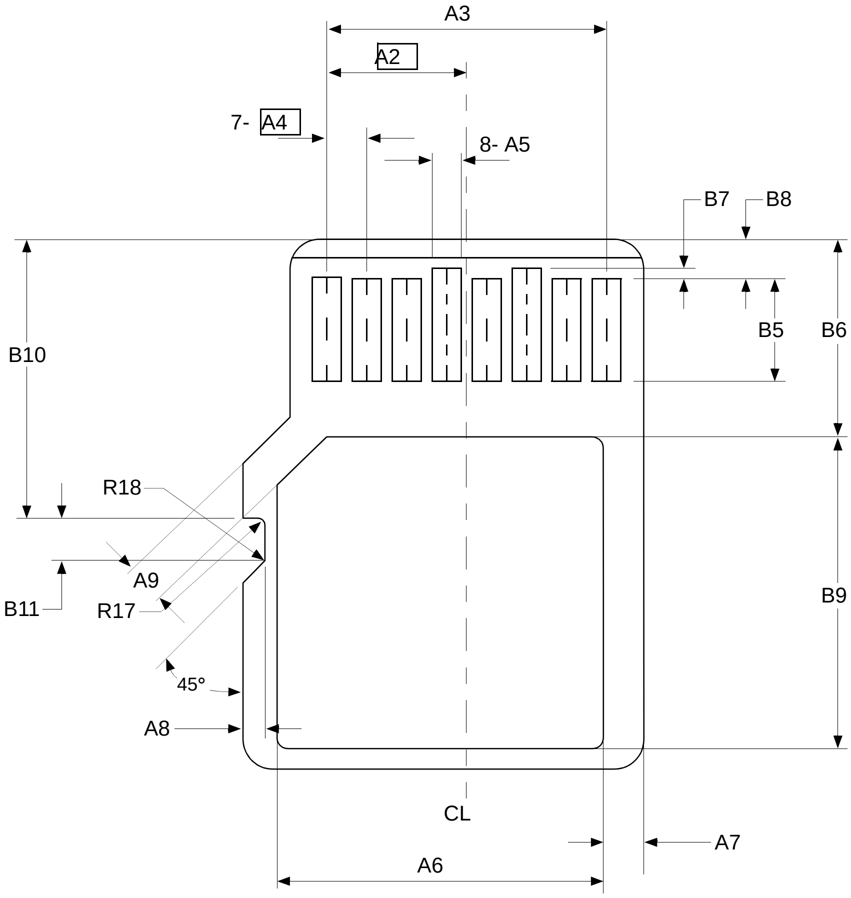


Figure 5-4. : Mechanical Description: Bottom View

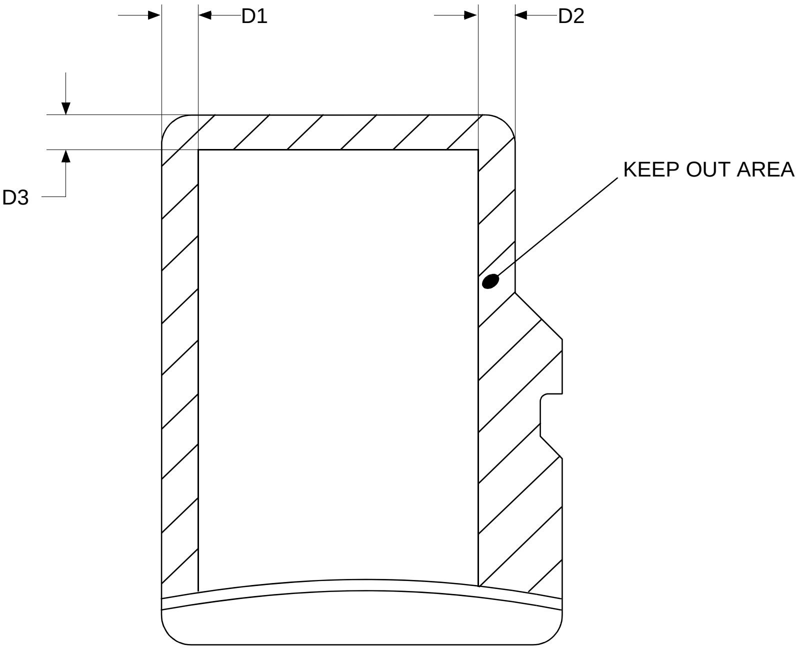


Figure 5-5. Mechanical Description: Keep Out Area

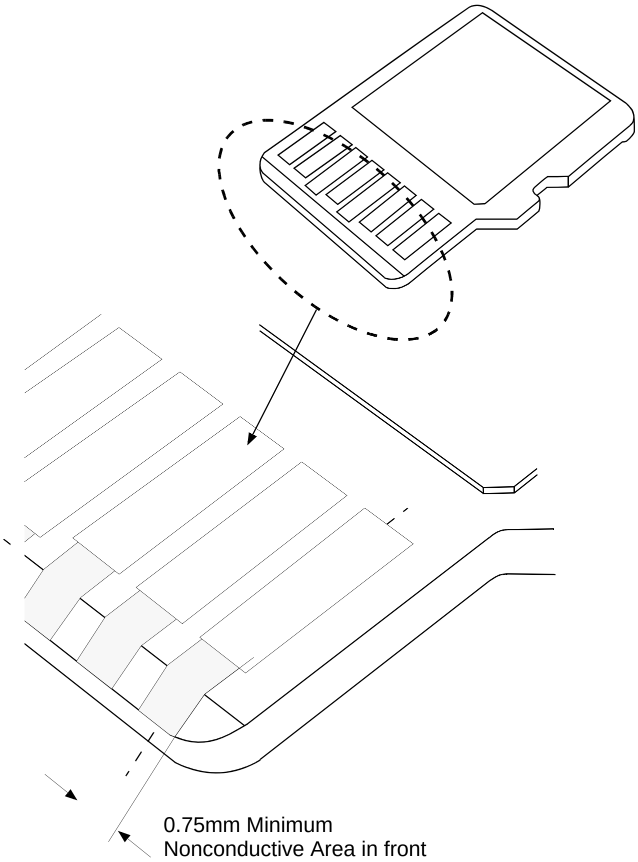


Figure 5-6. Nonconductive Area in Front of Contact Pad

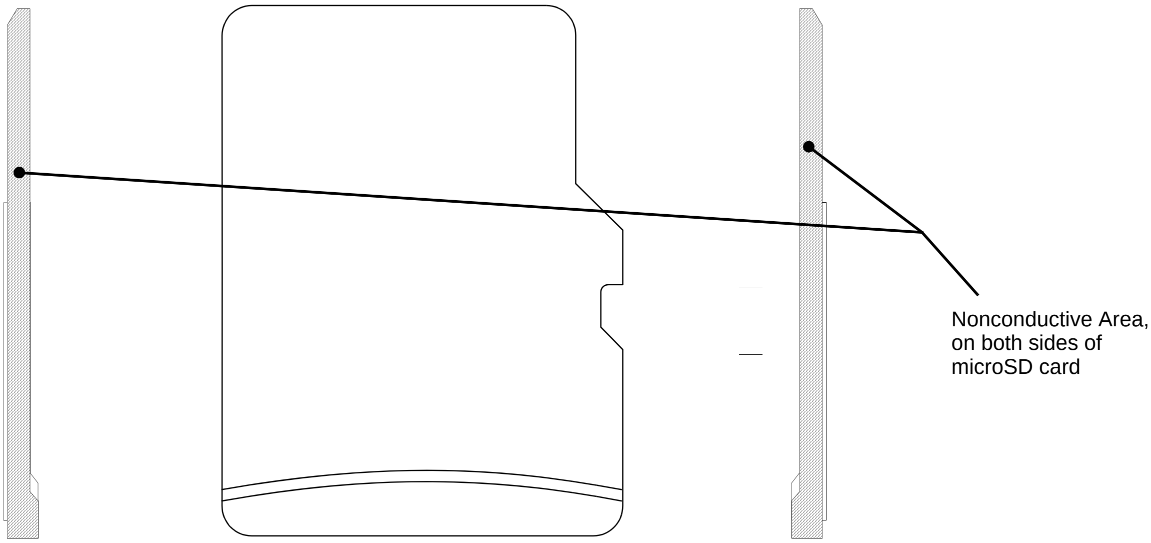


Figure 5-7. Nonconductive Area on Sides of Card

[Table 5-3] : microSD Package: Dimensions

	COMMON DIMENSIONS			
SYMBOL	MIN	NOM	MAX	NOTE
A	10.90	11.00	11.10	
A1	9.60	9.70	9.80	
A2	-	3.85	-	BASIC
A3	7.60	7.70	7.80	
A4	-	1.10	-	BASIC
A5	0.75	0.80	0.85	
A6	-	-	8.50	
A7	0.90	-	-	
A8	0.60	0.70	0.80	
A9	0.80	-	-	
B	14.90	15.00	15.10	
B1	6.30	6.40	6.50	
B2	1.64	1.84	2.04	
B3	1.30	1.50	1.70	
B4	0.42	0.52	0.62	
B5	2.80	2.90	3.00	
B6	5.50	-	-	
B7	0.20	0.30	0.40	
B8	1.00	1.10	1.20	
B9	-	-	9.00	
B10	7.80	7.90	8.00	
B11	1.10	1.20	1.30	
C	0.90	1.00	1.10	
C1	0.60	0.70	0.80	
C2	0.20	0.30	0.40	
C3	0.00	-	0.15	
D1	1.00	-	-	
D2	1.00	-	-	
D3	1.00	-	-	
R1	0.20	0.40	0.60	
R2	0.20	0.40	0.60	
R3	0.70	0.80	0.90	
R4	0.70	0.80	0.90	
R5	0.70	0.80	0.90	
R6	0.70	0.80	0.90	
R7	29.50	30.00	30.50	
R10	-	0.20	-	
R11	-	0.20	-	
R17	0.10	0.20	0.30	
R18	0.20	0.40	0.60	
R19	0.05	-	0.20	
R20	0.02	-	0.15	

NOTES:

1) DIMENSIONING AND TOLERTANCING PER ASME Y14.5M-1994.
2) DIMENSIONS ARE IN MILLIMETERS.
3) COPLANARITY IS ADDITIVE TO C1 MAX THICKNESS.

5.4.2 Mechanical Form Factor of SD Card

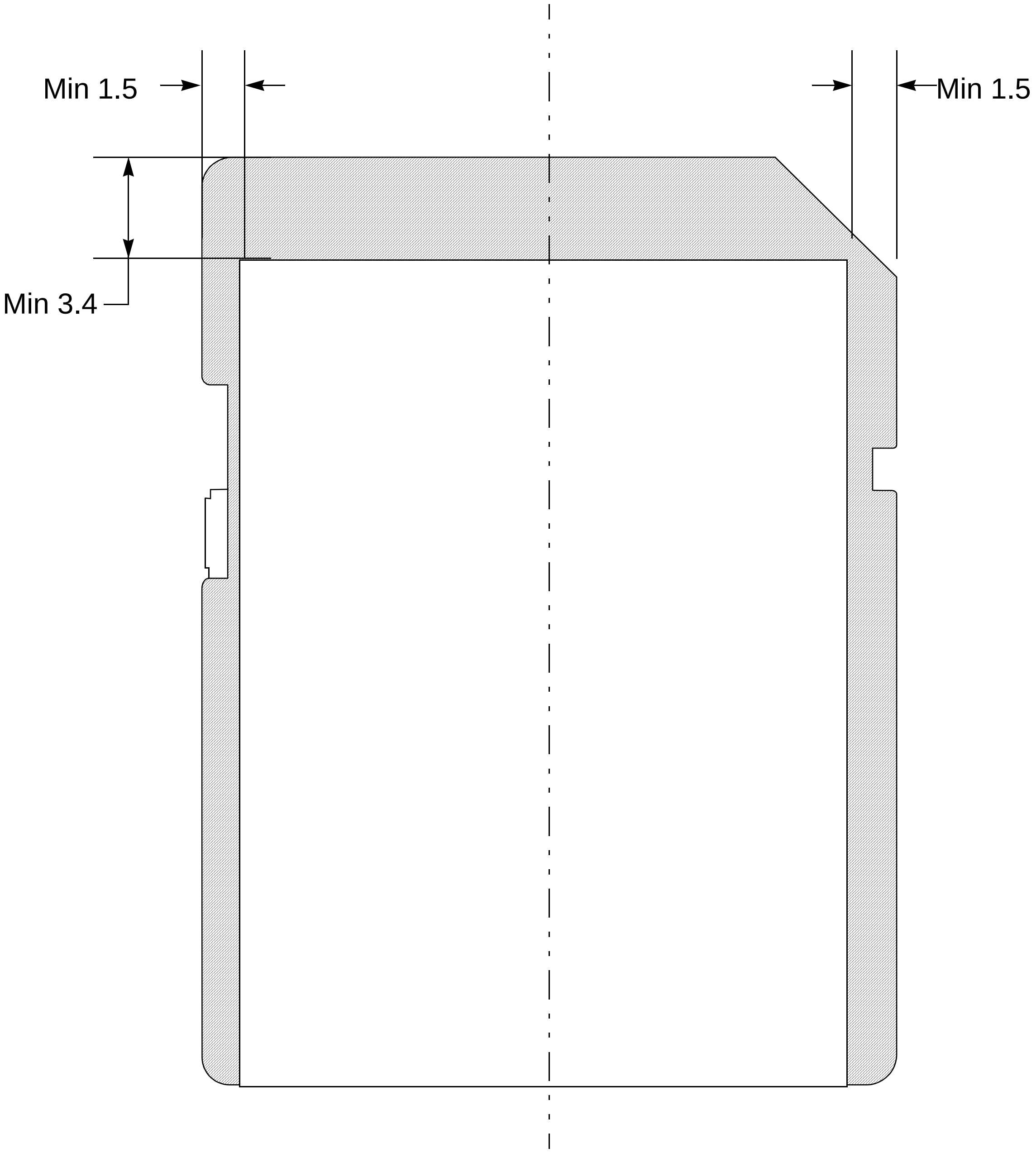


Figure 5-8. Mechanical Description: Top View - Keep Out Area

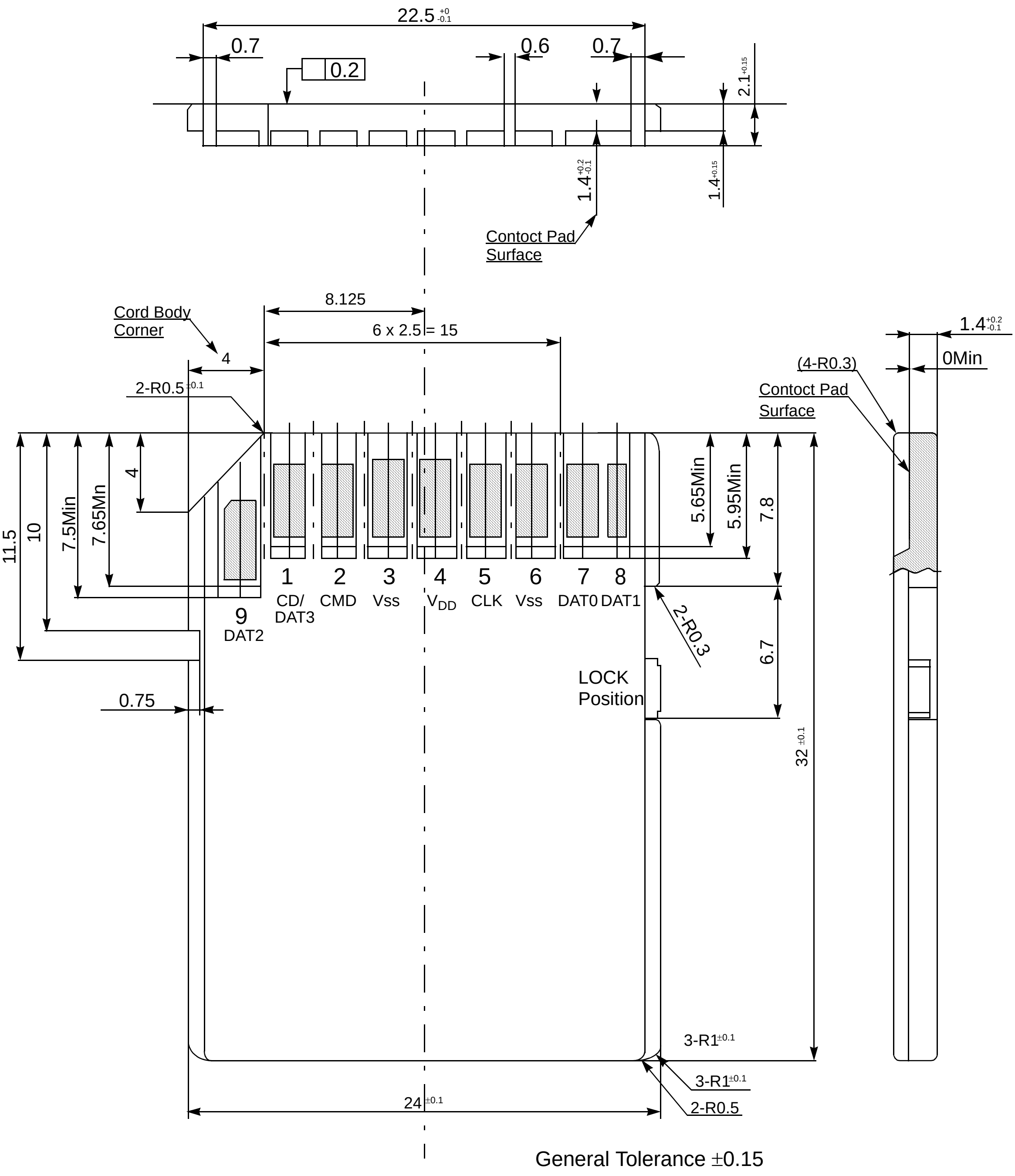


Figure 5-9. Mechanical Description

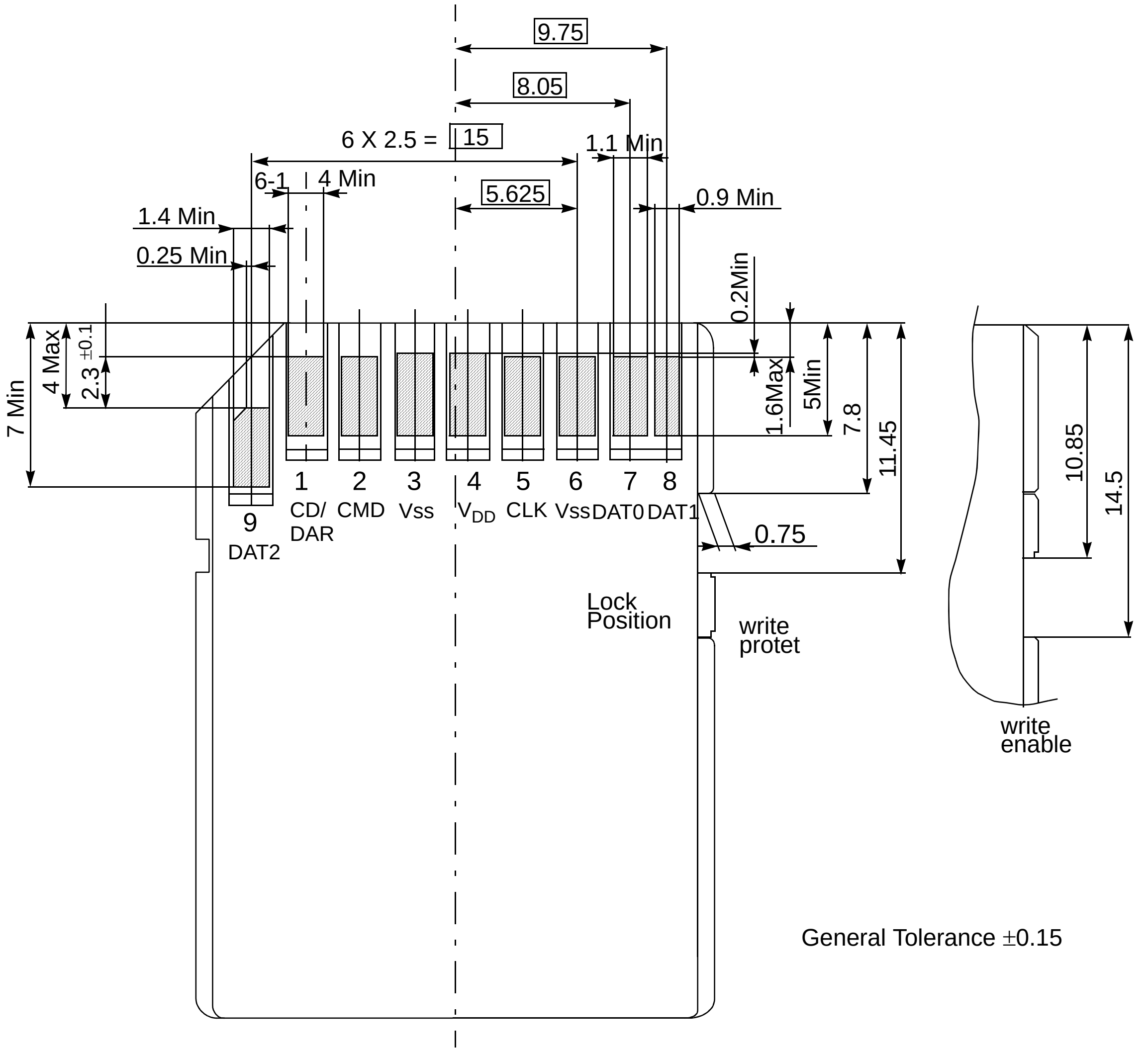


Figure 5-10. Mechanical Description: Bottom View

5.4.3 Electrical features, Environmental Reliability and Durability

SEC SD/microSD Card Electrical features, Environmental Reliabilities and Durabilities conform to SDA Physical Layer Specification Version 2.00, Section 8.1. For more details and informations of SEC SD/microSD Card Data, refer to Product Qualification Report.

5.5 Electrical Interface

The following sections provide valuable information about the electrical interface. See Chapter 6 of the SDA Physical Layer Specification, Version 3.00 for more detail information.

5.5.1 Power Up

The power-up of the SD/microSD Card bus is handled locally in each SD Card and in the host

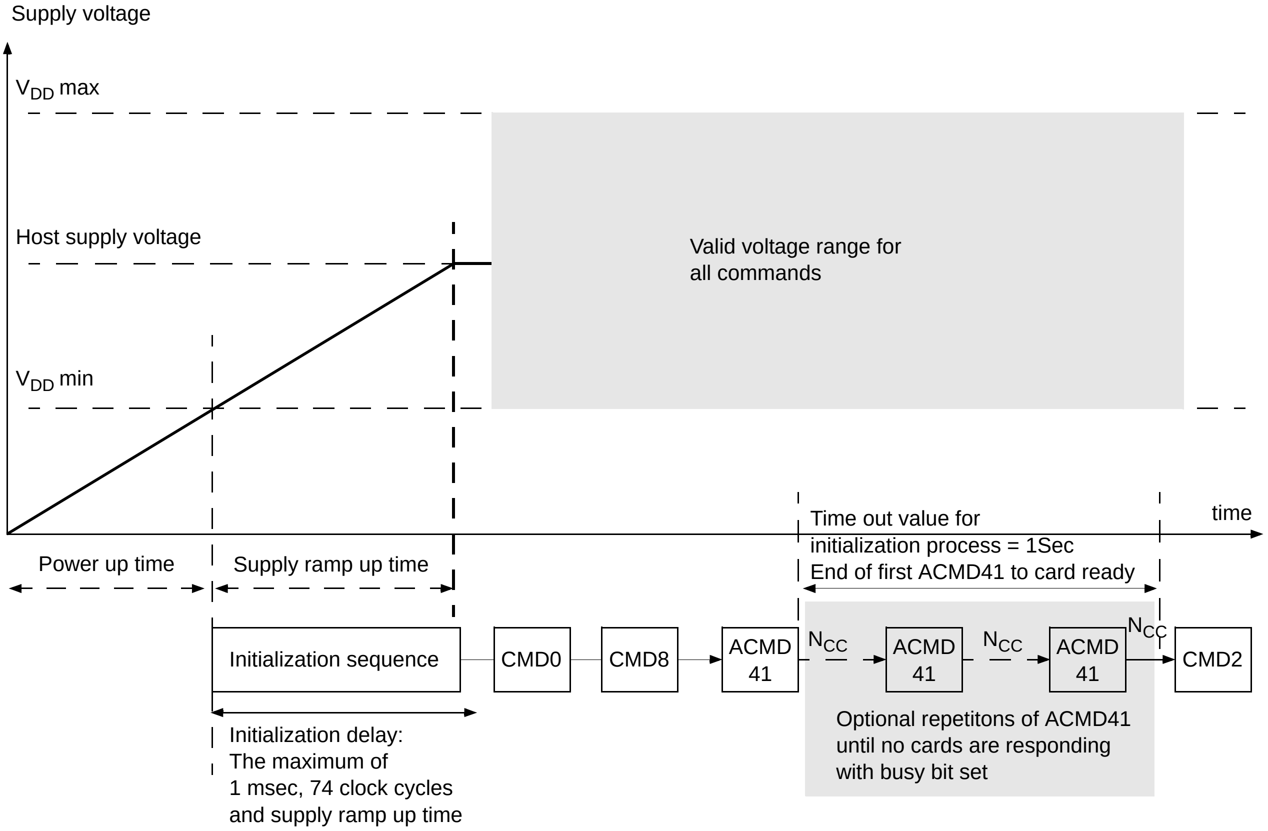


Figure 5-11. Power-up Diagram

.• Power up time is defined as voltage rising time from 0 volt to VDD(min.) and depends on application parameters such as the maximum number of SD Cards, the bus length and the characteristic of the power supply unit.

• Supply ramp up time provides the time that the power is built up to the operating level (the host supply voltage) and the time to wait until the SD card can accept the first command,

• The host shall supply power to the card so that the voltage is reached to VDD(min.) within 250ms and start to supply at least 74 SD clocks to the SD card with keeping CMD line to high. In case of SPI mode, CS shall be held to high during 74 clock cycles.

• After power up (including hot insertion, i.e. inserting a card when the bus is operating) the SD Card enters the *idle state*. In case of SD host, CMD0 is not necessary. In case of SPI host, CMD0 shall be the first command to send the card to SPI mode.

• CMD8 is newly added in the Physical Layer Specification Version 2.00 to support multiple voltage ranges and used to check whether the card supports supplied voltage. The version 2.00 host shall issue CMD8 and verify voltage before card initialization. The host that does not support CMD8 shall supply high voltage range.

• ACMD41 is a synchronization command used to negotiate the operation voltage range and to poll the cards until they are out of their power-up sequence. In case the host system connects multiple cards, the host shall check that all cards satisfy the supplied voltage. Otherwise, the host should select one of the cards and initialize.

5.5.2 Reset Level Power Up

Host needs to keep power line level less than 0.5V and more than 1ms before power ramp up.

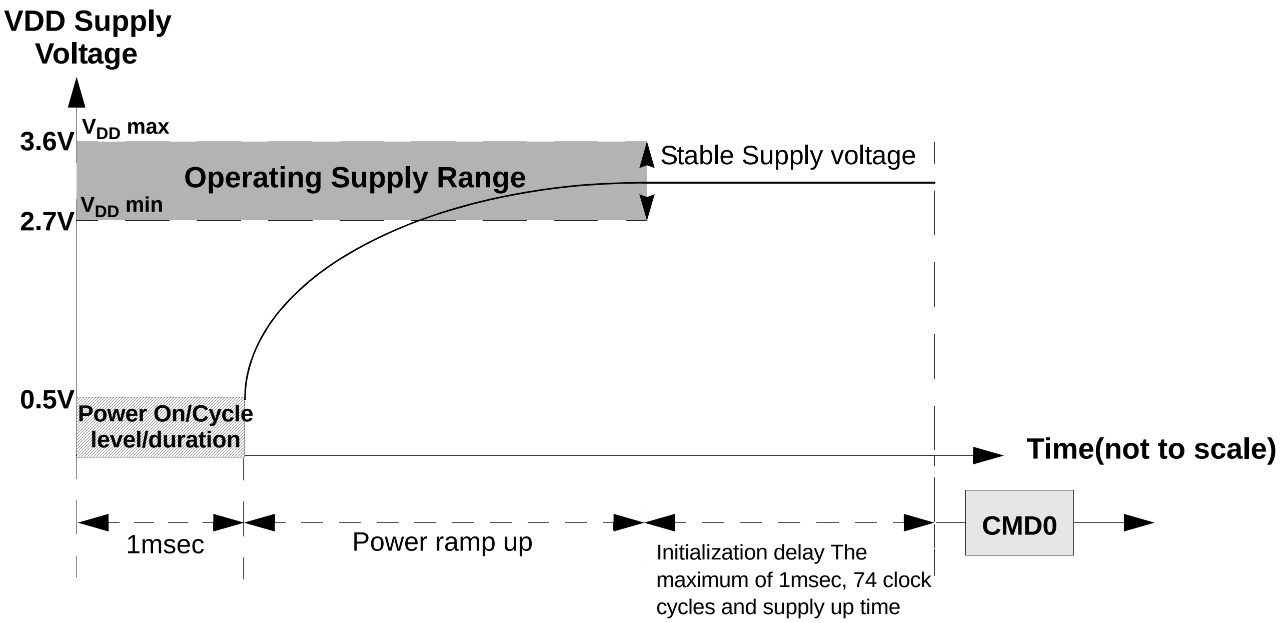


Figure 5-12. change of Figure for power up

- To assure a reliable SD Card hard reset of Power On and Power Cycle, Voltage level shall be below 0.5V and Time duration shall be at least 1ms.

- The power ramp up time is defined from 0.5V threshold level up to the operating supply voltage which is stable between VDD(min.) and VDD(max.) and host can supply SDCLK.

Followings are recommendation of Power ramp up:
(1) Voltage of power ramp up should be monotonic as much as possible.
(2) The minimum ramp up time should be 0.1ms.
(3) The maximum ramp up time should be 35ms for 2.7~3.6V power supply.

5.5.3 Power Down and Power Cycle

- When the host shuts down the power, the card VDD shall be lowered to less than 0.5Volt for a minimum period of 1ms. During power down, DAT, CMD, and CLK should be disconnected or driven to logical 0 by the host to avoid a situation that the operating current is drawn through the signal lines.

- If the host needs to change the operating voltage, a power cycle is required. Power cycle means the power is turned off and supplied again. Power cycle is also needed for accessing cards that are already in *Inactive State.* To create a power cycle the host shall follow the power down description before power up the card (i.e. the card VDD shall be once lowered to less than 0.5Volt for a minimum period of 1ms).

5.5.4 Bus Operating Conditions for 3.3V Signaling

SPI Mode bus operating conditions are identical to SD Card mode bus operating conditions.

5.5.4.1 Threshold Level for High Voltage Range

[Table 5-4] : Threshold Level for High Voltage

Parameter	Symbol	Min	Max.	Unit	Remark
Supply Voltage	V_{DD}	2.7	3.6	V	
Output High Voltage	V_{OH}	0.75*V_{DD}		V	I_{OH} = -2mA V_{DD} min
Output Low Voltage	V_{OL}		0.125*V_{DD}	V	I_{OL} = 2mA V_{DD} min
Input High Voltage	V_{IH}	0.625*V_{DD}	V_{DD}+0.3	V	
Input Low Voltage	V_{IL}	Vss-0.3	0.25 *V_{DD}	V	
Power Up Time	=		250	ms	From 0V to V_{DD} min

5.5.4.2 Bus Signal Line Load

The total capacitance of the SD Memory Card bus is the sum of the bus host capacitance CHOST, the bus capacitance CBUS itself and the capacitance CCARD of each card connected to this line:

Total bus capacitance = CHOST + CBUS + N * CCARD

Where N is the number of connected cards.

[Table 5-5] : Bus Operating Conditions - Signal Line's Load

Parameter	Symbol	Min	Max.	Unit	Remark
Pull-up resistance	R_{CMD} R_{DAT}	10	100	KOhm	to prevent bus floating
Total bus capacitance for each signal line	C_L		40	pF	1 card CHOST+CBUS shall not exceed 30 pF
Capacitance of the card for each siginal pin	C_{CARD}		10	pF	-
Maximum signal line inductance			16	nH	f_{PP} <= 20 MHz
Pull-up resistance inside card (pin1)	R_{DAT3}	10	90	KOhm	May be used for card detection
Capacity Connected to Power Line	CC		5	uF	To Prevent inrush current

Note that the total capacitance of CMD and DAT lines will be consist of CHOST, CBUS and one CCARD only because they are connected separately to the SD Memory Card host.
Host should consider total bus capacitance for each signal as the sum of CHOST, CBUS, and CCARD, these parameters are defined by per signal. The host can determine CHOST and CBUS so that total bus capacitance is less than the card estimated capacitance load (CL=40 pF). The SD Memory Card guarantees its bus timing when total bus capacitance is less than maximum value of CL (40 pF). To limit inrush current caused by host insertion, card maximum capacitance between VDD - VSS is defined as 5uF. To support host hot insertion, the host should consider decoupling capacitor connected to power line. As SD/microSD card Cc is 5uF(Max.), 45uF(min.) is recommended for Decoupling capacitor. For more details, please refer to Appendix E of the SDA Physical Layer Specification 3.00.

5.5.5 Bus Signal Levels

As the bus can be supplied with a variable supply voltage, all signal levels are related to the supply voltage.

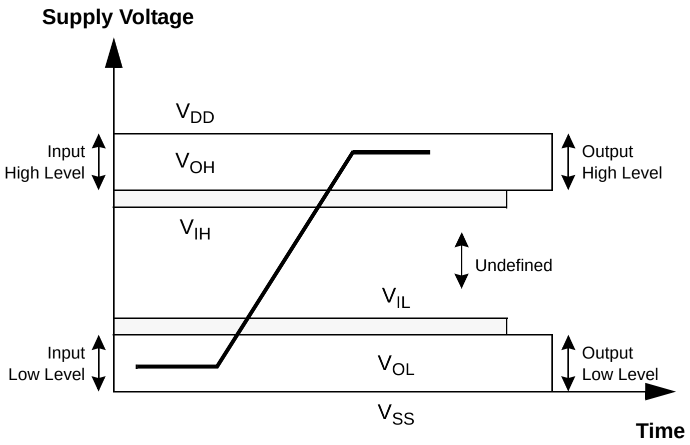


Figure 5-13. Bus Signal Levels

To meet the requirements of the JEDEC specification JESD8-1A and JESD8-7, the card input and output voltages shall be within the specified ranges shown in Table 5-2 for any VDD of the allowed voltage range.

5.5.6 Bus Timing (Default Mode)

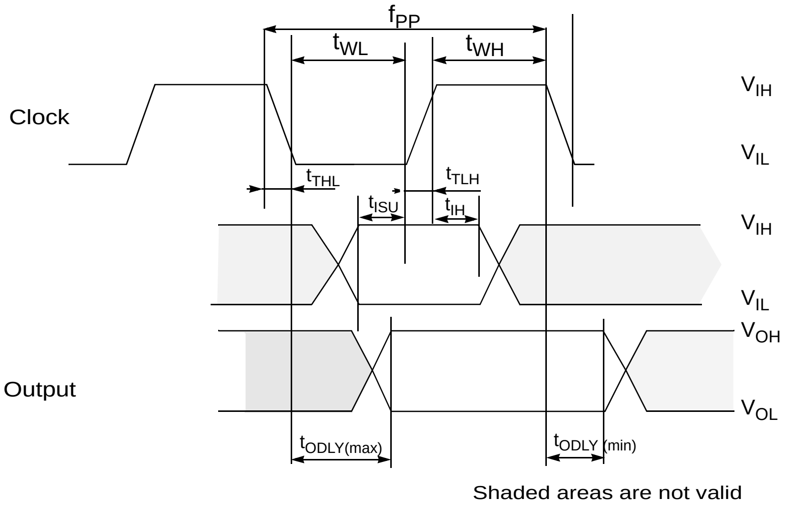


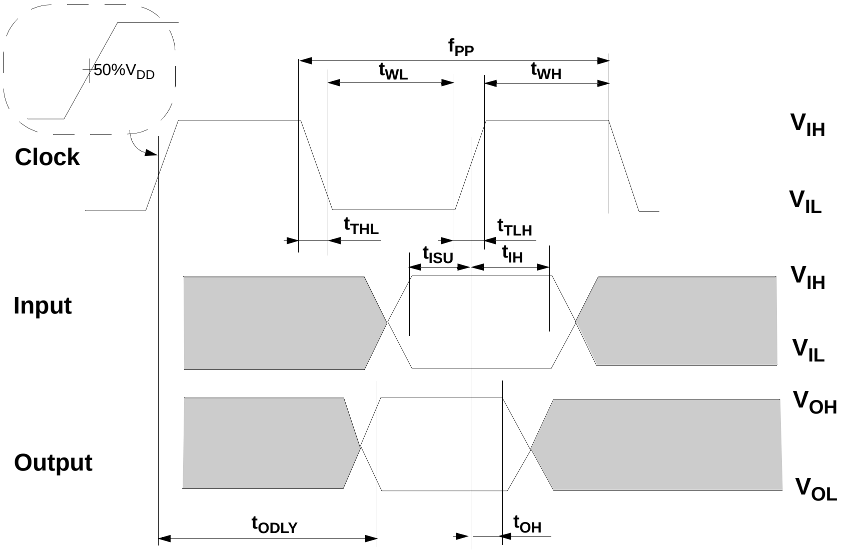
Figure 5-14. Timing diagram data input/output referenced to clock (Default)

[Table 5-6] : Bus Timing - Parameter Values (Default)

Parameter	Symbol	Min	Max.	Unit	Remark
Clock CLK (All values are referred to min. (V_{IH}) and max. (V_{IL})					
Clock frequency Data Transfer Mode	f_{PP}	0	25	MHz	C_{CARD} <= 10 pF (1 card)
Clock frequency Identification Mode	f_{OD}	0[1] / 100	400	kHz	C_{CARD} <= 10 pF (1 card)
Clock low time	t_{WL}	10		ns	C_{CARD} <= 10 pF (1 card's)
Clock high time	t_{WH}	10		ns	C_{CARD} <= 10 pF (1 card)
Clock rise time	t_{TLH}		10	ns	C_{CARD} <= 10 pF (1 card)
Clock fall time	t_{THL}		10	ns	C_{CARD} <= 10 pF (1 card)
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t_{ISU}	5		ns	C_{CARD} <= 10 pF (1 card)
Input hold time	t_{IH}	5		ns	C_{CARD} <= 10 pF (1 card)
Outputs CMD, DAT (referenced to CLK)					
Output delay time during Data Transfer Mode	t_{ODLY}	0	14	ns	C_L <= 40 pF (1 card)
Output delay time during Identification Mode	t_{ODLY}	0	50	ns	C_L<= 40 pF (1 card)

NOTE:
1) OHz means to stop the clock. The given minimum frequency range is for cases where a continuous clock is required

5.5.7 Bus Timing (High-speed Mode)



Shaded areas are not valid

Figure 5-15. Timing Diagram data Input/Output Refrenced to Clock (High-Speed)

[Table 5-7] : Bus Timing - Parameter Values (High-Speed)

Parameter	Symbol	Min	Max.	Unit	Remark
Clock CLK (All values are referred to min. (V_{IH}) and max. (V_{IL})					
Clock frequency Data Transfer Mode	f_{PP}	0	50	MHz	C_{CARD} <= 10 pF (1 card)
Clock low time	t_{WL}	7		ns	C_{CARD} <= 10 pF (1 card)
Clock high time	t_{WH}	7		ns	C_{CARD} <= 10 pF (1 card)
Clock rise time	t_{TLH}		3	ns	C_{CARD} <= 10 pF (1 card)
Clock fall time	t_{THL}		3	ns	C_{CARD} <= 10 pF (1 card)
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t_{ISU}	6		ns	C_{CARD} <= 10 pF (1 card)
Input hold time	t_{IH}	2		ns	C_{CARD} <= 10 pF (1 card)
Outputs CMD, DAT (referenced to CLK)					
Output delay time during Data Transfer Mode	t_{ODLY}		14	ns	C_L <= 40 pF (1 card)
Output Hold time	t_{OH}	2.5		ns	C_L <= 15 pF (1 card)
Total Systme capacitance for each line[1]	C_L		40	pF	1 card

NOTE:
1) In order to satisfy severe timing, host shall drive only one card.

6.0 SD/MICROSD CARD FUNCTIONAL DESCRIPTION

6.1 General

SEC SD/microSD Card Functional Description contained in this chapter; Section 6.2~6.14; basically, comfort to SDA Physical Layer Specification, Version 3.00. See Chapter 4 of the SDA Physical Layer Specification, Version 3.00 for detail information and guide.

6.2 Card Identification Mode

While in Card Identification mode the host resets all the cards that are in card identification mode, validates operation voltage range, identifies cards and asks them to publish Relative Card Address(RCA). This operation is done to each card separately on its own CMD line. Refer to Section 4.2 of the SDA Physical Layer Specification, Version 3.00 for detail information and guide[1)]

NOTE :
1) The products on this specification does not support UHS-1 mode. For correct identification flow, please refer to Section 4.2 of the SDA Physical Layer Specification, Version 2.00.

6.3 Clock Control

The SD/microSD Memory Card bus clock signal can be used by the host to change the cards to energy saving mode or to control the data flow(to avoid under-run or over-run conditions) on the bus. Refer to Section 4.4 of the SDA Physical Layer Specification, Version 3.00 for detail information and guide

6.4 Cyclic Redundancy Code

The CRC is intended for protecting SD Card commands, responses and data transfer against transmission errors on the SD Card bus. One CRC is generated for every command and checked for every response on the CMD line. For data blocks one CRC per transferred block, per data line, is generated. The CRC is generated and checked as described in the Section 4.5 of the SDA Physical Layer Specification, Version 3.0

6.5 Command

There are four kinds of commands defined to control the SD Card:

* Broadcast commands (bc), no response - The broadcast feature is only if all the CMD lines are connected together in the host. If they are separated then each card will accept it separately on his turn.
* Broadcast commands with response (bcr) - response from all cards simultaneously. Since there is no Open Drain mode in SD Card, this type of command is used only if all the CMD lines are separated. The command will be accepted and responded to by every card seperately.
* Addressed (point-to-point) commands (ac) - no data transfer on DAT lines
* Addressed (point-to-point) data transfer commands (adtc), data transfer on DAT lines

All commands and responses are sent over the CMD line of the SD Card bus. The command transmission always starts with the left bit of the bitstring corresponding to the command code word. For more details, refer to the Section 4.7 of the SDA Physical Layer Specification, Version 3.0.

NOTE:
Limited Vendor CMD information, only for certain customer and application, can be provided under appropriate purpose of usage.

6.6 Memory Array Partitioning

The basic unit of data transfer to/from the SD Card is one byte. All data transfer operations which require a block size always define block lengths as integer multiples of bytes. Some special functions need other partition granularity.

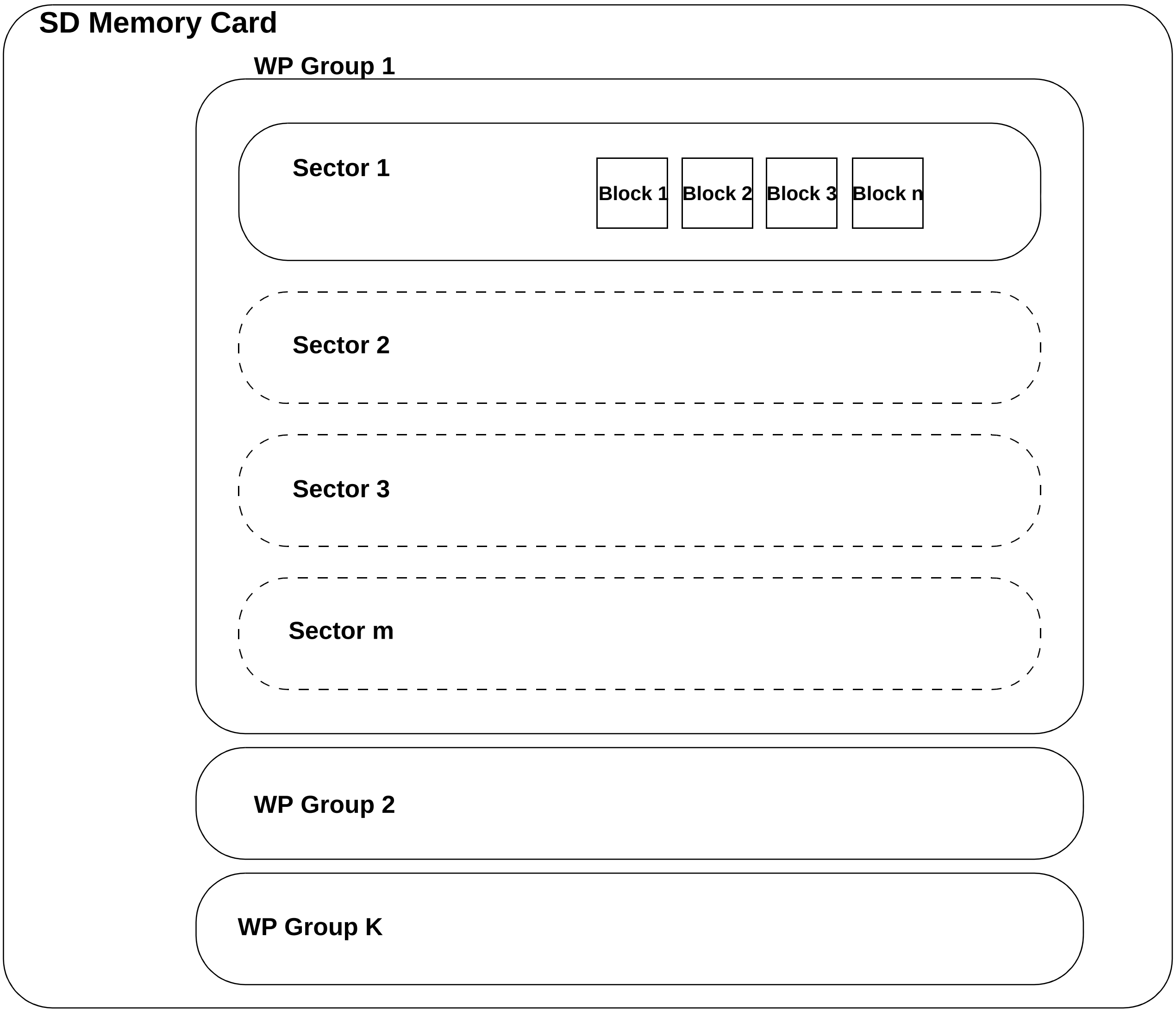


Figure 6-1: Write Protection Hierarchy

For block oriented commands, the following definition is used:

- Block: is the unit that is related to the block oriented read and write commands. Its size is the number of bytes that will be transferred when one block command is sent by the host. The size of a block is either programmable or fixed. The information about allowed block sizes and the programmability is stored in the CSD.
- For devices that have erasable memory cells, special erase commands are defined. The granularity of the erasable units is in general not the same as for the block oriented commands:
- Sector: is the unit that is related to the erase commands. Its size is the number of blocks that will be erased in one portion. The size of a sector is fixed for each device. The information about the sector size (in blocks) is stored in the CSD. Note that if the card specifies AU size, sector size should be ignored.
- AU (Allocation Unit): is a physical boundary of the card and consists of one or more blocks and its size depends on each card. The maximum AU size is defined for memory capacity. Furthermore AU is the minimal unit in which the card guarantees its performance for devices which complies with Speed Class Specification. The information about the size and the Speed Class are stored in the SD Status. AU is also used to calculate the erase timeout.
- WP-Group: is the minimal unit that may have individual write protection for devices which support write-protected group. Its size is the number of groups that will be write-protected by one bit. The size of a WP-group is fixed for each device. The information about the size is stored in the CSD.
  The High Capacity SD Memory Card does not support the write protect group command.

6.7 Timings

Refer to Section 4.12 of the SDA Physical Layer Specification, Version 3.00 for detail information and guide[1)]

NOTE :
1) The products on this specification does not support UHS-1 mode.

6.8 Speed Class Specification

Refer to Section 4.13 of the SDA Physical Layer Specification, Version 3.00 for detail information and guide[1)]

NOTE :
1)The products on this specification does not support UHS-1 mode.

6.9 Erase Timeout Calculation

Refer to Section 4.14 of the SDA Physical Layer Specification, Version 3.00 for detail information and guide[1)]

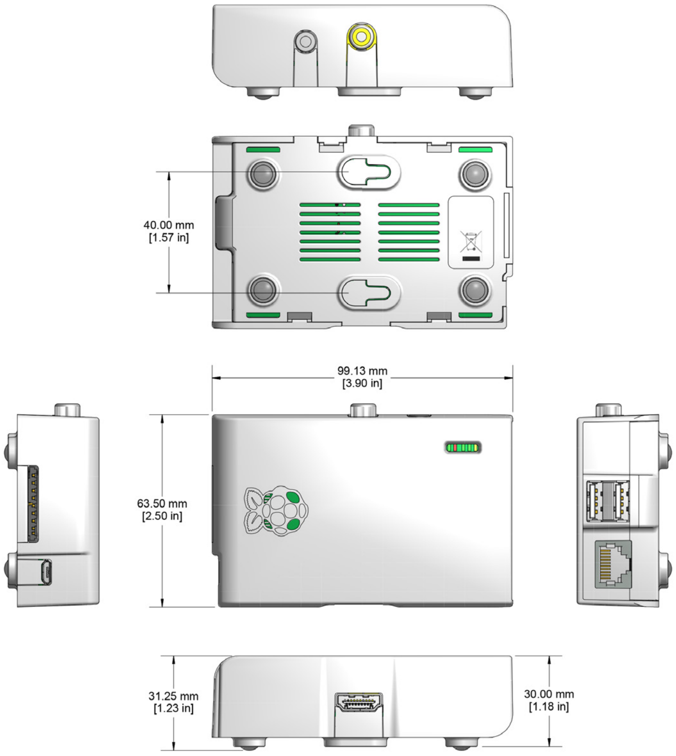
NOTE :
1) The products on this specification does not support UHS-1 mode.

Raspberry Pi Enclosures

multicomp

Features

- Two piece construction, base and cover, moulded in a smooth finish ABS plastic
- Base has side clips which positively locate the Raspberry Pi board into position allowing it to be operated with or without the cover in place, giving easy access to the GPIO and JTAG
- Base comes with four rubber feet and also features two screw slots to allow the enclosure to be vertically mounted
- Cover can be easily removed without the need to unplug any cables
- The cover features a tunnel recess to allow viewing of the operation LEDs
- Vented base and cover to allow through air flow
- Available in White, Black or Clear



Dimensions : Millimeters (Inches)

Part Number Table

Description	Part Number
Enclosure, Raspberry Pi, White	MC-RP001-WHT
Enclosure, Raspberry Pi, Black	MC-RP001-BLK
Enclosure, Raspberry Pi, Clear	MC-RP001-CLR

Important Notice : This data sheet and its contents (the "Information") belong to the members of the Premier Farnell group of companies (the "Group") or are licensed to it. No licence is granted for the use of it other than for information purposes in connection with the products to which it relates. No licence of any intellectual property rights is granted. The Information is subject to change without notice and replaces all data sheets previously supplied. The Information supplied is believed to be accurate but the Group assumes no responsibility for its accuracy or completeness, any error in or omission from it or for any use made of it. Users of this data sheet should check for themselves the Information and the suitability of the products for their purpose and not make any assumptions based on information included or omitted. Liability for loss or damage resulting from any reliance on the Information or use of it (including liability resulting from negligence or where the Group was aware of the possibility of such loss or damage arising) is excluded. This will not operate to limit or restrict the Group's liability for death or personal injury resulting from its negligence. Multicomp is the registered trademark of the Group. © Premier Farnell plc 2012.

USB Adaptor

pro-POWER

Specifications

Input Voltage	: 100 to 240V AC ~ 50/60Hz
Output Voltage	: 5V DC
Output Current	: 1,000mA
Input Method	: Plug-In
Function	: USB output

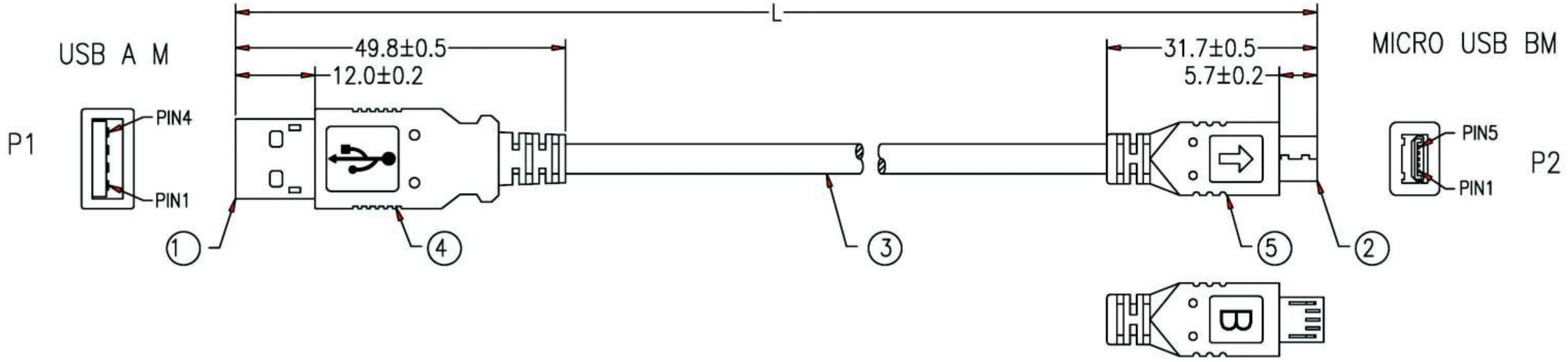
Part Number Table

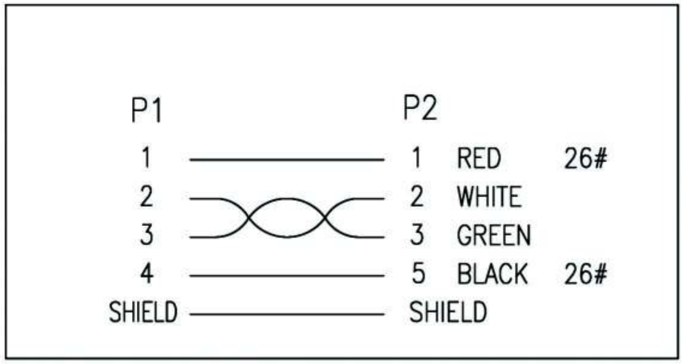
Description	Part Number
Adaptor, USB to UK, Black	MWUSBUK1 BLACK
Adaptor, USB to UK, White	MWUSBUK1

Important Notice : This data sheet and its contents (the "Information") belong to the members of the Premier Farnell group of companies (the "Group") or are licensed to it. No licence is granted for the use of it other than for information purposes in connection with the products to which it relates. No licence of any intellectual property rights is granted. The Information is subject to change without notice and replaces all data sheets previously supplied. The Information supplied is believed to be accurate but the Group assumes no responsibility for its accuracy or completeness, any error in or omission from it or for any use made of it. Users of this data sheet should check for themselves the Information and the suitability of the products for their purpose and not make any assumptions based on information included or omitted. Liability for loss or damage resulting from any reliance on the Information or use of it (including liability resulting from negligence or where the Group was aware of the possibility of such loss or damage arising) is excluded. This will not operate to limit or restrict the Group's liability for death or personal injury resulting from its negligence. pro-POWER is the registered trademark of the Group. © Premier Farnell plc 2012.

P1:	M-942A
P2:	M-Y668A M-668A

Customer No:	No.43





NOTE: All relevant materials are complied with RoHS directive .

No.	DESCRIPTION	Q'TY	Supplier
5	MOLDED PE COLOR TRANSPARENT		M280
	MOLDED PVC BLACK		M304
4	MOLDED PVC BLACK		M304
3	USB 2.0 CABLE, UL STYLE 1 TWISTED PAIR 28AWG AND 2C 26AWG, AMA BRAID SHIELDING WITH DRAIN WIRE, OD 4.5mm, JACKET BLACK	L-20mm	M304
2	MICRO USB B MALE 2.0, GOLD PLATING 30u" CONTACTS, BLACK INSULATOR	1	M108
1	USB A MALE 2.0, GOLD PLATING 30u" CONTACTS, WHITE INSULATOR, SHORT TYPE	1	M108

APPROVED		USB2-160
CHECKED		
DRAWING		
DATE	23/JAN/09	ITEM NO.

Cable Length
Unless Other Spec.
Tolerances:

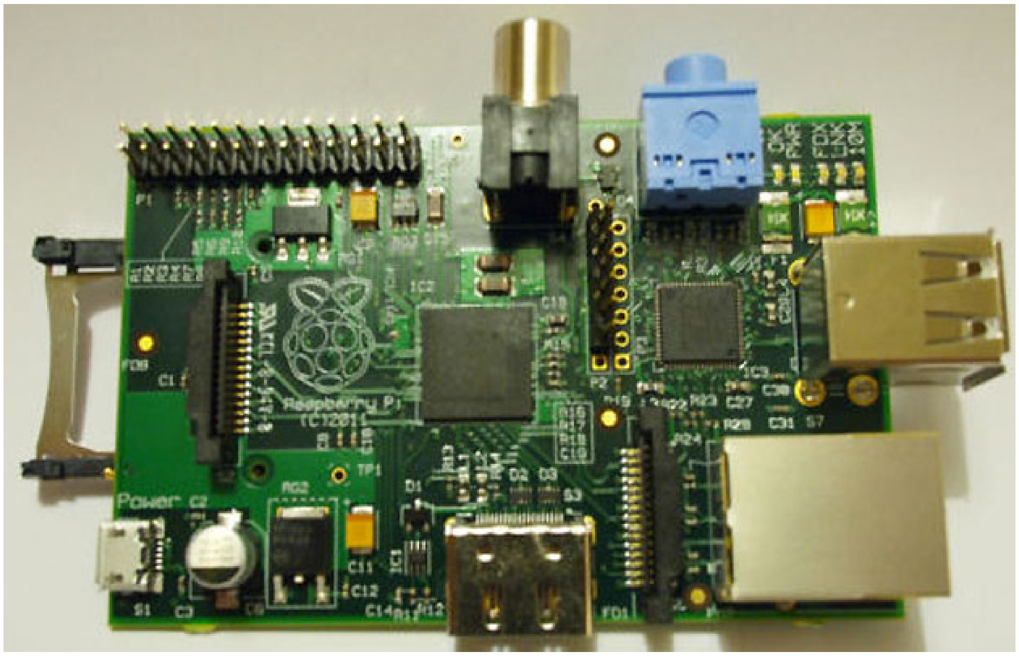
0 ~200mm	±10mm
201~1000mm	±20mm
1001 ~1500mm	±25mm
1501~2100mm	±30mm
2100mm↑	±40mm

UNIT:mm

CHECKED

Quick Start Guide

The Raspberry Pi – Single Board Computer



Source: Raspberry Pi & Wiki

Chapter 1: RPi Hardware Basic Setup

Typical Hardware You Will Need

While the RPi can be used without any additional hardware (except perhaps a power supply of some kind), it won't be much use as a general computer. As with any normal PC, it is likely you will need some additional hardware.

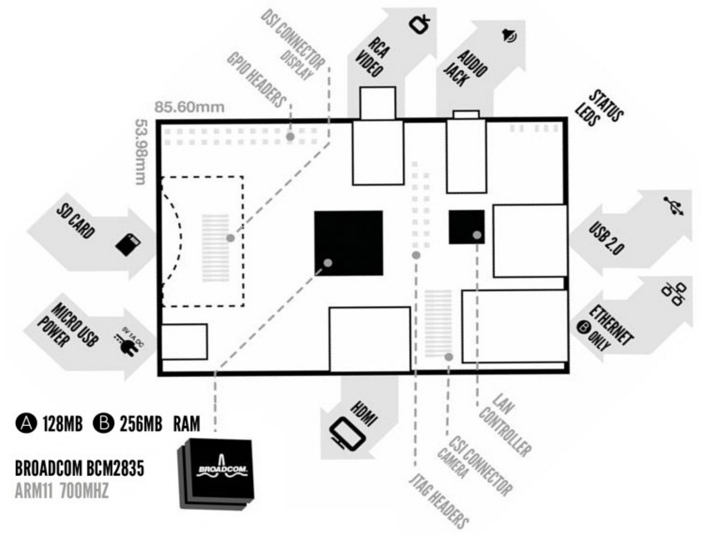
The following are more or less essential:

- Raspberry Pi board
- Prepared Operating System SD Card
- USB keyboard
- Display (with HDMI, DVI, Composite or SCART input)
- Power Supply
- Cables

Highly suggested extras include:

- USB mouse
- Internet connectivity - a USB WiFi adaptor (Model A/B) or a LAN cable (Model B)
- Powered USB Hub
- Case

Connecting Together



You can use the diagram to connect everything together, or use the following instructions:

1. Plug the preloaded SD Card into the Pi.
2. Plug the USB keyboard and mouse into the Pi, perhaps via a USB Hub. Connect the Hub to power, if necessary.

3. Plug the video cable into the screen (TV) and into the Pi.
4. Plug your extras into the Pi (USB WiFi, Ethernet cable, hard drive etc.). This is where you may really need a USB Hub.
5. Ensure that your USB Hub (if any) and screen are working.
6. Plug the power source into the main socket.
7. With your screen on, plug the other end of the power source into the Pi.
8. The Pi should boot up and display messages on the screen.

It is always recommended to connect the MicroUSB Power to the unit last (while most connections can be made live, it is best practice to connect items such as displays/h/w pin connections with the power turned off).

The RPi may take a long time to boot when powered-on for the first time, so be patient!

Prepared Operating System SD Card

As the RPi has no internal storage or built-in operating system it requires an SD-Card that is set up to boot the RPi.

- You can create your own preloaded card using any suitable SD card you have. Be sure to backup any existing data on the card.
- Preloaded SD cards will be available from the RPi Shop.

This guide will assume you have a preloaded SD card.

Keyboard & Mouse

Most standard USB keyboards and mice will work with the RPi. Wireless keyboard/mice should also function, and only require a single USB port for an RF dongle. In order to use a Bluetooth keyboard or mouse you would need to use a Bluetooth dongle, which again uses a single port.

Remember that the Model A has a single USB port and the Model B only has two (typically a keyboard and mouse will use a USB port each).

Display

There are two main connection options for the RPi display, *HDMI* (high definition) and *Composite* (low definition).

- HD TVs and most LCD Monitors can be connected using a full-size 'male' HDMI cable, and with an inexpensive adaptor if DVI is used. HDMI versions 1.3 and 1.4 are supported, and a version 1.4 cable is recommended. The RPi outputs audio and video via HMDI, but does not support HDMI input.
- Older TVs can be connected using Composite (a yellow-to-yellow cable) or via SCART (using a Composite to SCART adaptor). PAL and NTSC TVs are supported. When using composite video, audio is available from a 3.5mm (1/8 inch) socket, and can be sent to your TV, to headphones, or to an amplifier. To send audio your TV,

you will need a cable which adapts from 3.5mm to double (red and white) RCA connectors.

Note: There is no VGA output available, so older VGA monitors will require an expensive adaptor.

Using an HDMI to DVI-D (digital) adaptor plus a DVI to VGA adaptor will not work. HDMI does not supply the DVI-A (analogue) needed to convert to VGA - converting an HDMI or DVI-D source to VGA (or component) needs an active converter. (It can work out cheaper to buy a new monitor.) The lack of VGA has been acknowledged as a priority issue.

Power Supply

The unit uses a Micro USB connection to power itself (only the power pins are connected - so it will not transfer data over this connection). A standard modern phone charger with a micro-USB connector will do, but needs to produce at least 700mA at 5 volts. Check your power supply's ratings carefully. Suitable mains adaptors will be available from the RPi Shop and are recommended if you are unsure what to use.

You can use a range of other power sources (assuming they are able to provide enough current ~700mA):

- Computer USB Port or powered USB hub (will depend on power output)
- Special wall warts with USB ports
- Mobile Phone Backup Battery (will depend on power output) (in theory - needs confirmation)

To use the above, you'll need a USB A 'male' to USB micro 'male' cable - these are often shipped as data cables with MP3 players.

Cables

You will probably need a number of cables in order to connect your RPi up.

1. Micro-B USB Power Cable
2. HDMI-A or Composite cable, plus DVI adaptor or SCART adaptor if required, to connect your RPi to the Display/Monitor/TV of your choice.
3. Audio cable, this is not needed if you use a HDMI TV/monitor.
4. Ethernet/LAN Cable

Additional Peripherals

You may decide you want to use various other devices with your RPi, such as Flash Drives/Portable Hard Drives, Speakers etc.

Internet Connectivity

This may be an Ethernet/LAN cable (standard RJ45 connector) or a USB WiFi adaptor. The RPi ethernet port is auto-sensing which means that it may be connected to a router or directly to another computer (without the need for a crossover cable).

USB-Hub

In order to connect additional devices to the RPi, you may want to obtain a USB Hub, which will allow multiple devices to be used.

It is recommended that a **powered** hub is used - this will provide any additional power to the devices without affecting the RPi itself.

USB version 2.0 is recommended. USB version 1.1 is fine for keyboards and mice, but may not be fast enough for other accessories.

Case

Since the RPi is supplied without a case, it will be important to ensure that you do not use it in places where it will come into contact with conductive metal or liquids, unless suitably protected.

Expansion & Low Level Peripherals

If you plan on making use of the low level interfaces available on the RPi, then ensure you have suitable header pins for the GPIO (and if required JTAG) suitable for your needs.

Also if you have a particular low-level project in mind, then ensure you design in suitable protection circuits to keep your RPi safe.

Chapter 2: RPi Advanced Setup

Finding hardware and setting up

You'll need a preloaded SD card, USB keyboard, TV/Monitor (with HDMI/ DVI/ Composite/ SCART input), and power supply (USB charger or a USB port from a powered USB Hub or another computer).

You'll likely also want a USB mouse, a case, and a USB Hub (a necessity for Model A). A powered USB Hub will reduce the demand on the RPi. To connect to the Internet, you'll need either an Ethernet/LAN cable (Model B) or a USB WiFi adaptor (either model).

When setting up, it is advisable to connect the power after everything else is ready.

Serial connection

The Serial Port is a simple and uncomplicated method to connect to the Raspberry Pi. The communication depends on byte wise data transmission, is easy to setup and is generally available even before boot time.

First interaction with the board

Connect the serial cable to the COM port in the Raspberry Pi, and connect the other end to the COM port or USB Serial Adapter in the computer.

Serial Parameters

The following parameters are needed to connect to the Raspberry. All parameters except **Port_Name** and **Speed** are default values and may not need to be set.

- **Port_Name**: Linux automatically assigns different names for different types of serial connectors. Choose your option:
  - Standard Serial Port: ttyS0 ... ttySn
  - USB Serial Port Adapter: ttyUSB0 ... ttyUSBn
- **Speed**: 115200
- Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None

The Serial Port is generally usable by the users in the group **dialout**. To add oneself to the group **dialout** the the following command needs to be executed with **root** privileges:

```
$useradd -G {dialout} your_name
```

- **Super Easy Way Using GNU Screen**

Enter the command below into a terminal window

```
screen Port_Name 115200
```

- **Super Easy Way Using Minicom**

Run minicom with the following parameters:

```
minicom -b 115200 -o -D Port_Name
```

- **GUI method with GtkTerm**

Start *GtkTerm*, select Configuration->Port and enter the values above in the labelled fields.

- **Windows Users**

Windows Users above Windows XP must download putty or a comparable terminal program. Users of XP and below can choose between using *putty* and *Hyperterminal.*

First Dialog

If you get the prompt below, you are connected to the Raspberry Pi shell!

```
prompt> #
```

First command you might want try is "help":

```
prompt> # help
```

If you get some output, you are correctly connected to the Raspberry Pi! Congratulations!

SD card setup

Now we want to use an SD card to install some GNU/Linux distro in it and get more space for our stuff. You can use either an SD or SDHC card. In the latter case of course take care that your PC card reader also supports SDHC. Be aware that you are not dealing with an x86 processor, but instead a completely different architecture called ARM, so don't forget to install the ARM port for the distro you are planning to use.

Formatting the SD card via the mkcard.txt script

1. Download **mkcard.txt** .
2. $ chmod +x mkcard.txt

3. $ ./mkcard.txt /dev/sd*x*, where *x* is the letter of the card. You can find this by inserting your card and then running dmesg | tail. You should see the messages about the device being mounted in the log. Mine mounts as **sdc**.

Once run, your card should be formatted.

Formatting the SD card via fdisk "Expert mode"

First, lets clear the partition table:

```
===========================================================================
=====
$ sudo fdisk /dev/sdb

Command (m for help): o
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by
w(rite)
===========================================================================
=====
```

Print card info:

```
===========================================================================
=====
Command (m for help): p

Disk /dev/sdb: 128 MB, 128450560 bytes
....
===========================================================================
=====
```

Note card size in bytes. Needed later below.

Then go into "Expert mode":

```
===========================================================================
=====
Command (m for help): x
===========================================================================
=====
```

Now we want to set the geometry to 255 heads, 63 sectors and calculate the number of cylinders required for the particular SD/MMC card:

```
===========================================================================
=====
Expert command (m for help): h
Number of heads (1-256, default 4): 255

Expert command (m for help): s
Number of sectors (1-63, default 62): 63
Warning: setting sector offset for DOS compatiblity
===========================================================================
=====
```

NOTE: Be especially careful in the next step. First calculate the number of cylinders as follows:

- B = Card size in bytes (you got it before, in the second step when you printed the info out)
- C = Number of cylinders

```
C=B/255/63/512
```

When you get the number, you round it DOWN. Thus, if you got 108.8 you'll be using 108 cylinders.

```
===========================================================================
=====
Expert command (m for help): c
Number of cylinders (1-1048576, default 1011): 15
===========================================================================
=====
```

In this case 128MB card is used (reported as 128450560 bytes by fdisk above), thus 128450560 / 255 / 63 / 512 = 15.6 rounded down to 15 cylinders. Numbers there are 255 heads, 63 sectors, 512 bytes per sector.

So far so good, now we want to create two partitions. One for the boot image, one for our distro. Create the FAT32 partition for booting and transferring files from Windows. Mark it as bootable.

```
===========================================================================
=====
Expert command (m for help): r
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
```

```
First cylinder (1-245, default 1): (press Enter)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-245, default 245): +50

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): c
Changed system type of partition 1 to c (W95 FAT32 (LBA))

Command (m for help): a
Partition number (1-4): 1
===========================================================================
=====
```

Create the Linux partition for the root file system.

```
===========================================================================
=====
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (52-245, default 52): (press Enter)
Using default value 52
Last cylinder or +size or +sizeM or +sizeK (52-245, default 245):(press
Enter)
Using default value 245
===========================================================================
=====
```

Print and save the new partition records.

```
===========================================================================
=====
Command (m for help): p

Disk /dev/sdc: 2021 MB, 2021654528 bytes
255 heads, 63 sectors/track, 245 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

  Device Boot      Start         End      Blocks   Id  System
/dev/sdc1   *           1          51      409626    c  W95 FAT32 (LBA)
/dev/sdc2              52         245     1558305   83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

```
WARNING: Re-reading the partition table failed with error 16: Device or
resource busy. The kernel still uses the old table. The new table will be
used at the next reboot.

WARNING: If you have created or modified any DOS 6.x partitions, please see
the fdisk manual page for additional information.
Syncing disks.
===========================================================================
=====
```

Now we've got both partitions, next step is formatting them.

NOTE: If the partitions (/dev/sdc1 and /dev/sdc2) does not exist, you should unplug the card and plug it back in. Linux will now be able to detect the new partitions.

```
===========================================================================
=====
$ sudo mkfs.msdos -F 32 /dev/sdc1 -n LABEL
mkfs.msdos 2.11 (12 Mar 2005)

$ sudo mkfs.ext3 /dev/sdc2
mke2fs 1.40-WIP (14-Nov-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
195072 inodes, 389576 blocks
19478 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=402653184
12 block groups
32768 blocks per group, 32768 fragments per group
16256 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information:
===========================================================================
=====
```

All done!

NOTE: For convenience, you can add the -L option to the mkfs.ext3 command to assign a volume label to the new ext3 filesystem. If you do that, the new (automatic) mount point under /media when you insert that SD card into some Linux hosts will be based on that label. If there's no label, the new mount point will most likely be a long hex string, so assigning a label makes manual mounting on the host more convenient.

Setting up the boot partition

The boot partition must contain:

- bootcode.bin : 2nd stage bootloader, starts with SDRAM disabled
- loader.bin : 3rd stage bootloader, starts with SDRAM enabled
- start.elf: The GPU binary firmware image, provided by the foundation.
- kernel.img: The OS kernel to load on the ARM processor. Normally this is Linux - see instructions for compiling a kernel.
- cmdline.txt: Parameters passed to the kernel on boot.

Optional files:

- config.txt: A configuration file read by the GPU. Use this to override set the video mode, alter system clock speeds, voltages, etc.
- vlls directory: Additional GPU code, e.g. extra codec's. Not present in the initial release.

Additional files supplied by the foundation

These files are also present on the SD cards supplied by the foundation.

Additional kernels. Rename over kernel.img to use them (ensure you have a backup of the original kernel.img first!):

- kernel_emergency.img : kernel with busybox rootfs. You can use this to repair the main linux partition using e2fsck if the linux partition gets corrupted.

Additional GPU firmware images, rename over start.elf to use them:

- arm128_start.elf : 128M ARM, 128M GPU split (use this for heavy 3D work, possibly also required for some video decoding)
- arm192_start.elf : 192M ARM, 64M GPU split (this is the default)
- arm224_start.elf : 224M ARM, 32M GPU split (use this for Linux only with no 3D or video processing. It's enough for the 1080p frame buffer, but not much else)

Writing the image into the SDcard and finally booting GNU/Linux

The easiest way to do this is to use PiCard. It even saves you from some hassles explained above. You will need your SD card + reader and a Linux pc to use PiCard. After that, just plug the card into your Rpi.

Setting up the boot args

Wire up your Raspberry Pi and power it up

As explained in Chapter 1

SD Card Cloning/Backup

```
Note: Update these instructions if required once they've been tried
```

From windows you can copy the full SD-Card by using Win32DiskImager. Alternatively, you can use the following instructions;

```
Note:
Many built-in SD card readers do not work, so if you have problems
use an external SD-USB adapter for this.
```

Required Software Setup

- download a windows utility dd.exe from *http://www.chrysocome.net/dd*
- rename it windd.exe

(This executable can to write to your harddisk so exercise caution using it!)

- make a copy named dd-removable.exe

(That executable refuses to write to your hard disk as it is named dd-removable As long as you use dd-removable.exe you cannot lose your hard disk)

- Connect an SD card to the computer
- run "dd-removable –list"

Should give something like this:

```
rawwrite dd for windows version 0.6beta3.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by terms of the GPL Version 2.

NT Block Device Objects
\\?\Device\Harddisk1\Partition0
link to \\?\Device\Harddisk1\DR8
Removable media other than floppy. Block size = 512
size is 4075290624 bytes
```

This "\\?\Device\Harddisk1\Partition0" is the part you need.

Reading an image from the SD Card

BEWARE: DO THIS WRONG AND YOU CAN LOSE YOUR HARDDISK!!!

Obviously, you can NOT use 'dd-removable' to read an image as that executable refuses to write to your hard disk (so extra care is required here as you use 'windd').

- To **read** an SD-card image from the SD-card use:

```
windd bs=1M if=\\?\Device\Harddisk1\Partition0 of=THE_IMAGE_READ -size
Your disk name ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Copying an image to the SD Card

BEWARE: DO THIS WRONG AND YOU CAN LOSE YOUR HARDDISK!!!

- To **copy** an image named "THEIMAGE" to the SD-card do this:

```
dd-removable bs=1M if=THEIMAGE of=\\?\Device\Harddisk1\Partition0
                    Your disk name ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Software Development/Proving

A supported platform for the Raspberry is Qt , which is already being worked on. C/C++ is supported through a gcc cross-compiling tool chain.

After compiling, using QEMU and a Linux VM would be one way of testing your apps. This also works on Windows. Search the forum for the readymade ARM images. The choice of programming languages, IDEs and other tools ON the R-Pi is only determined by:

- The operating system compatibility (at the moment the specific Linux distro used)
- The status of the respective ARM package repositories and their binary compatibility
- The possibility to build other software + its dependencies for the R-Pi from sources.

For more guides and projects involving the Raspberry Pi, see RPi Projects (http://elinux.org/RPi_Projects).