

## SD card component

---

REV A

Publication Date: 2013/12/16  
XMOS © 2013, All Rights Reserved.



## Table of Contents

<b>1 Overview</b>	<b>3</b>
1.1 Features	3
1.2 Memory requirements	3
1.3 Resource requirements	3
1.4 Performance	3
<b>2 Hardware requirements</b>	<b>4</b>
2.1 Recommended hardware	4
2.1.1 sliceKIT	4
2.2 Demonstration applications	4
2.2.1 SD Card test application	4
<b>3 API</b>	<b>5</b>
3.1 Configuration defines	5
3.2 API	5
<b>4 Programming guide</b>	<b>7</b>
4.1 Source code structure	7
4.2 Executing the project	7
4.3 Software requirements	7
<b>5 Example applications</b>	<b>8</b>
5.1 app_sdcard_test	8
5.2 Application notes	8
5.2.1 Getting started	8
5.2.2 Import and Build the Application	8
5.2.3 Run the Application	9

# 1 Overview

---

## IN THIS CHAPTER

- ▶ Features
  - ▶ Memory requirements
  - ▶ Resource requirements
  - ▶ Performance
- 

The SD card module is used to talk to the SD card through the SPI interface.

## 1.1 Features

- ▶ Low level functions for accessing SD card
- ▶ Uses SPI interface

## 1.2 Memory requirements

Resource	Usage
Stack	220 bytes
Program	5 KB

## 1.3 Resource requirements

Resource	Usage
Clocks	1
Threads	1

## 1.4 Performance

The achievable effective bandwidth varies according to the available xCORE MIPS. The maximum pixel clock supported is 25MHz.

## 2 Hardware requirements

---

### IN THIS CHAPTER

- ▶ Recommended hardware
  - ▶ Demonstration applications
- 

## 2.1 Recommended hardware

### 2.1.1 sliceKIT

This module may be evaluated using the sliceKIT modular development platform, available from digikey. Required board SKUs are:

- ▶ XP-SKC-L2 (sliceKIT L2 Core Board)
- ▶ XA-SK-XTAG2 (sliceKIT xTAG adaptor)
- ▶ XA-SK-FLASH 1V0 Slice Card

## 2.2 Demonstration applications

### 2.2.1 SD Card test application

- ▶ Package: `sc_sdcard`
- ▶ Application: `app_sdcard_test`

This demo uses the `module_FatFs` along with the `module_sdcardSPI`, `module_spi_master`.

Required board SKUs for this demo are:

- ▶ XP-SKC-L16 (sliceKIT L16 Core Board) plus XA-SK-XTAG2 (sliceKIT xTAG adaptor)

## 3 API

---

### IN THIS CHAPTER

- ▶ Configuration defines
  - ▶ API
- 

- ▶ module: module\_spi\_master

The below section details the APIs in the SD card module. For details about the FatFS APIs please refer to the respective repositories.

### 3.1 Configuration defines

The `module_sdcardSPI` requires a configuration defined in `spi_conf.h`. The module requires nothing to be additionally defined.

#### **SPI\_MASTER\_SD\_CARD\_COMPAT**

This defines the SPI modifications needed for SD card usage. This needs to be set to 1 for SD cards to use SPI interface.

### 3.2 API

- ▶ `SDCardHostSPI.xc`
- ▶ `spi_conf.h`

The SD card module provides APIs to read/write data to SD card.

The SD card APIs are as follows:

`DSTATUS disk_initialize(BYTE drv)`

Function to initialize disk.

This function has the following parameters:

`drv` Physical drive number (0).

`DSTATUS disk_status(BYTE drv)`

Function to open or create a file.

This function has the following parameters:

`drv` Drive number (always 0).

DRESULT disk\_read(BYTE drv, BYTE buff[], DWORD sector, BYTE count)

Function to read data from a disk.

This function has the following parameters:

drv	Physical drive number (0).
buff	Pointer to the data buffer to store read data.
sector	Start sector number (LBA).
count	Sector count (1..128).

DRESULT disk\_write(BYTE drv, const BYTE buff[], DWORD sector, BYTE count)

Function to write data to the disk.

This function has the following parameters:

drv	Physical drive number (0).
buff	Pointer to the data to be written.
sector	Start sector number (LBA).
count	Sector count (1..128).

DRESULT disk\_ioctl(BYTE drv, BYTE ctrl, BYTE buff[])

Function to control device specific features and miscellaneous functions.

This function has the following parameters:

drv	Physical drive number (0).
ctrl	Control code.
buff	Buffer to send/receive control data.

The SD card APIs use the module\_spi\_master APIs.

## 4 Programming guide

---

### IN THIS CHAPTER

- ▶ Source code structure
  - ▶ Executing the project
  - ▶ Software requirements
- 

### 4.1 Source code structure

### 4.2 Executing the project

### 4.3 Software requirements

The module is built on xTIMEcomposer version 13.0. The module can be used in version 13.0 or any higher version of xTIMEcomposer.

## 5 Example applications

---

### IN THIS CHAPTER

- ▶ `app_sdcard_test`
  - ▶ Application notes
- 

This tutorial describes a demo application that reads and writes files on SD card using the SPI interface.

### 5.1 `app_sdcard_test`

- ▶ Demonstrate read and write files on SD card using the SPI interface. At the end, it prints the read/write performances of the filesystem on SD card using the SPI.

### 5.2 Application notes

#### 5.2.1 Getting started

1. Connect XA-SK-FLASH 1V0 Slice Card to the XP-SKC-L16 sliceKIT Core board using the connector marked with the TRIANGLE.
2. Connect the xTAG Adapter to sliceKIT Core board, and connect xTAG-2 to the adapter.
3. Connect the xTAG-2 to host PC. Note that the USB cable is not provided with the sliceKIT starter kit.
4. Set the XMOS LINK to OFF on the xTAG Adapter(XA-SK-XTAG2).
5. Make sure the SD card slot in XA-SK-FLASH slice has a Class-4 SD card in it.
6. Switch on the power supply to the sliceKIT Core board.

#### 5.2.2 Import and Build the Application

1. Open xTIMEcomposer and check that it is operating in online mode. Open the edit perspective (Window->Open Perspective->XMOS Edit).
2. Locate the 'SD card demo' item in the xSOFTip pane on the bottom left of the window and drag it into the Project Explorer window in the xTIMEcomposer. This will also cause the modules on which this application depends to be imported as well.



3. Click on the `app_sdcard_test` item in the Explorer pane then click on the build icon (hammer) in xTIMEcomposer. Check the console window to verify that the application has built successfully.

### 5.2.3 Run the Application

Now that the application has been compiled, the next step is to run it on the sliceKIT Core Board using the tools to load the application over JTAG (via the xTAG-2 and xTAG Adapter card) into the xCORE multicore microcontroller.

1. Select the file `app_sdcard_test.xe` in the `app_sdcard_test` project from the Project Explorer.
2. Click on the Run icon (the white arrow in the green circle).
3. At the Select Device dialog select XMOS xTAG-2 connect to L1[0..1] and click OK.
4. The application starts executing and reads/writes contents into SD card.



Copyright © 2013, All Rights Reserved.

---

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.

XMOS and the XMOS logo are registered trademarks of Xmos Ltd. in the United Kingdom and other countries, and may not be used without written permission. All other trademarks are property of their respective owners. Where those designations appear in this book, and XMOS was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.