Application Report
SLAA527–March 2012

Creating Autoplay Installers Using MSP430F5529 and CDFS

Timothy Logan *MSP430 Applications*

ABSTRACT

This application report explains the implementation of using the CDFS feature of the Mass Storage Controller (MSC) API of the USB Developer's Package for the MSP430™ Microcontroller.

Utilizing Compact Disc File System (CDFS) as a file system on the storage medium allows a mass storage device to be recognized by a host as a standard read-only CD-ROM device. This allows software vendors to bundle installers for their products on a MSP430 controlled storage device while taking advantage of features such as autoplay capabilities and write-protected content. The ISO 9660 file format is used in combination with the File Allocation Table (FAT) file system to provide a quick and easy way to virtually "mount" an ISO image onto a MSP430 controlled device.

This application report demonstrates the use of the MSC stack using a MSP430F5529 Experimenter's Board. The Experimenter Board's standard Secure Digital card is used as the storage medium. The storage backend is arbitrary and can be changed to match the end application. A SPI flash chip, for example, could also be used to store the installer data.

Source code and other information discussed in this application report can be downloaded from http://www.ti.com/lit/zip/slaa527.

Contents

List of Figures

1 Introduction

In this software implementation, the FatFS implementation of the File Allocation Table (FAT) file system is used for performing FAT reads. FatFS is a third party implementation of FAT that is targeted for use with embedded architectures such as the MSP430 microcontrollers. Any write requests sent to the MSP430 are ignored when dealing with CDFS; because the storage medium is presented to the host as a read-only CD device, write operations to the device are not allowed. The write protection is handled on the operating system layer and does not concern the MSP430. Recording to recordable disc media is currently not supported by the TI USB Stack.

This implementation also takes advantage of premade ISO 9660 (.iso) files as a mechanism for mounting virtual CD devices. This method of CD emulation is operating system agnostic. Most CD-ROM and storage devices use a standard and generic set of storage commands called SCSI commands. These commands are a standard and are issued and understood by the majority of operating systems (Mac OSX, Windows, Linux). The MSP430, more specifically the MSC portion of the USB Stack, is responsible for receiving SCSI commands from the host, decoding them, and taking the appropriate actions based on the intent of the command. Many freeware applications exist for creating ISO files from physical CD mediums. The creation and authoring of ISO files is not described in this application note.

2 Experimenter's Board Implementation

The MSP430F5529 USB Experimenter's Board (MSP-EXP430F5529) is equipped with an SD card reader. Any standard microSD card can be used and is compatible with the TI USB Development APIs. The FatFS libraries are used for initialization, setup, and management of the SD card. These libraries provide an easy and convenient method of managing the SD card without having to access low-level register calls.

The first portion of this tutorial describes how to install a file system on the SD card so that an ISO file can be copied and mounted. In the attached source package, open the M5_Example-FatFS example. This example is identical to the M5_Example-FatFS that is included in the MSP430 USB Developer's Package. The code example presents a raw SCSI volume to the host with a size as large as the inserted SD card. It is up to the host to format the card with the correct file system. After building and programming the binary to the experimenter's board, the host operating (in the case of this example, that is Windows 7) should recognize the drive and prompt you to format. Choose to format with the FAT32 file system (see Figure 1).

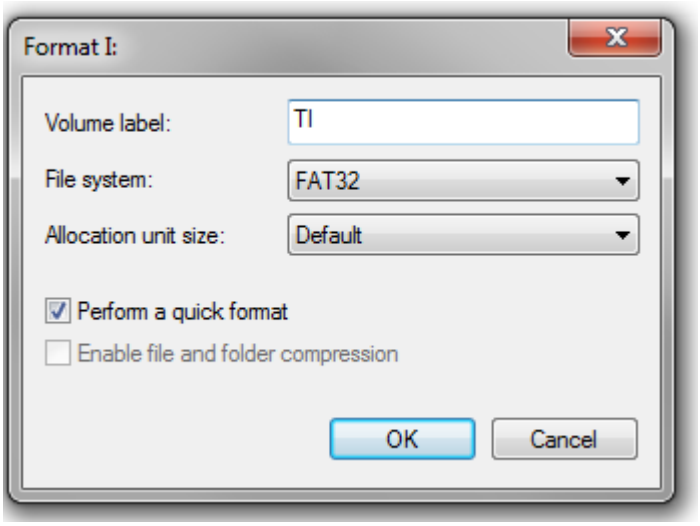

Figure 1. Formatting the SD Card

After the FAT32 file system is formatted on the SD card, copy the ISO file to the SD card and rename it to mount.iso (see Figure 2).

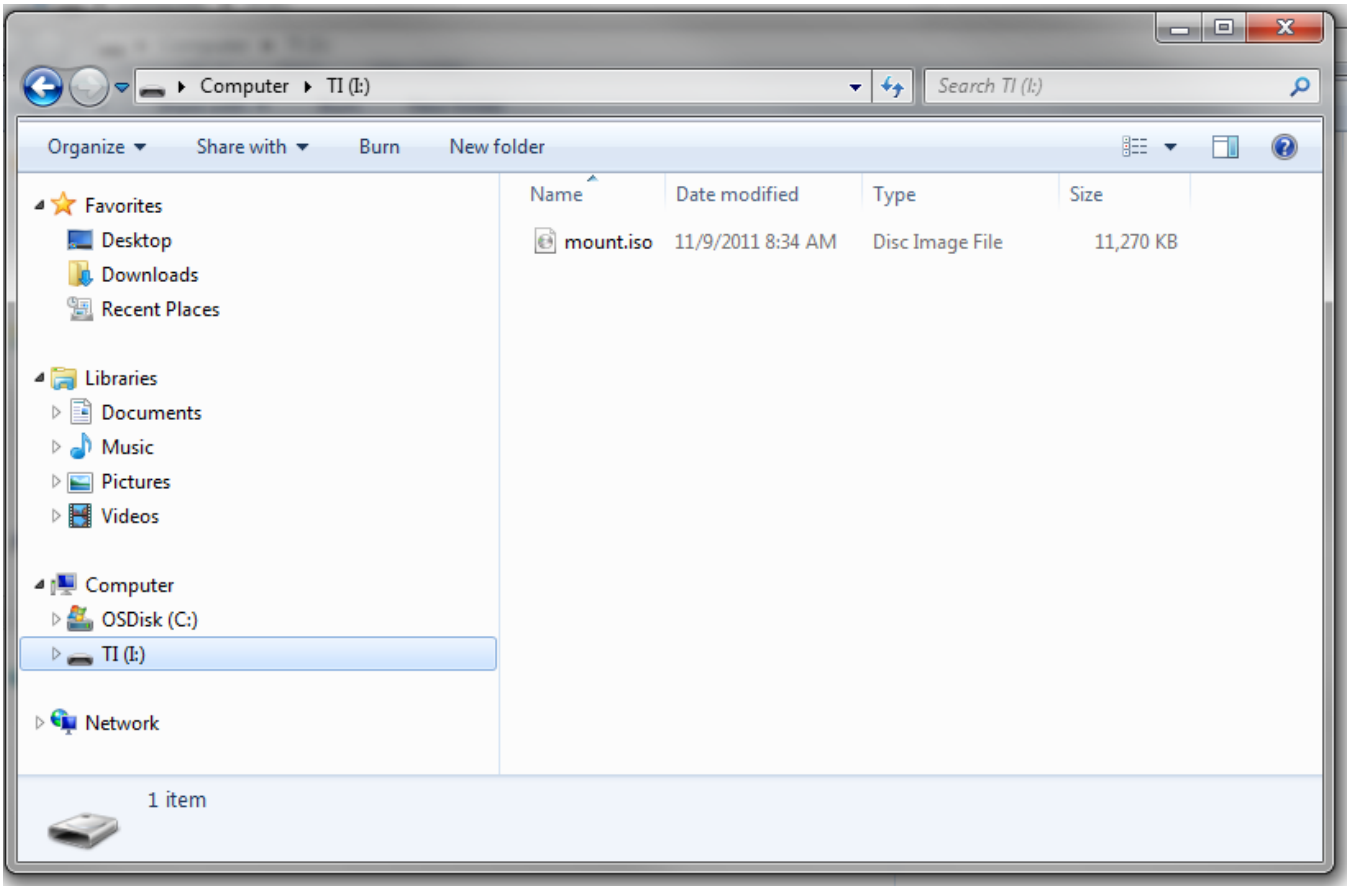

Figure 2. ISO File

The next part of this example is to create an application that dynamically reads the ISO on the FAT32 file system and presents it to the host as a CD-ROM device. First, open the M5_Example-CDFS project in the attached source package. The key part of the code that enables CD-ROM support is the define macro within the descriptors.h file in the USB_Config directory of the example.

```
#define CDROM_SUPPORT
```

This define statement tells the USB MSC API to include all of the necessary code and overhead to enable CDFS support. Any mass storage device presented to the host is reported as a CD-ROM device. Appropriate USB descriptors can also be generated automatically using the USB Descriptor Tool program that is included within the USB Developer's Package. Before implementing read commands, the following initializations commands must be issued to tell FatFS which file to read.

```
f_mount(0, &fs);
fres = f_open(&isoFile, "0:mount.iso", FA_OPEN_EXISTING | FA_READ);
```

This code mounts the file system to the corresponding file system handle and opens mount.iso in the root directory of the file system. After the file is opened, the application needs to direct read accesses to the ISO file during runtime. In the example, find the part of the code that handles SCSI read requests sent from the host:

```
while (RWbuf_info->operation == kUSBMSC_READ)
```

Within this while loop, the application translates host read requests to the appropriate address in the ISO file structure. Because the CD_ROMSUPPORT macro was defined, the host is expecting a CD file structure. The main part of the code that manages CD reads is as follows:

```
f_lseek(&isoFile, RWbuf_info->lba * BYTES_PER_BLOCK);
fres = f_read(&isoFile, RWbuf_info->bufferAddr, RWbuf_info->lbCount * BYTES_PER_BLOCK, &br);
```

The f_lseek call is responsible for setting the correct offset in the ISO file for the read operation. It accepts a file handle and an offset parameter in bytes. The SCSI read operation from the host provides a read address in the form of a logical block address (LBA). To calculate the byte position in the ISO file to read, the LBA is multiplied by the bytes per block of the file system (in this case 512). After seeking to the appropriate address in the file, the requested data is read, checked to make sure the read was successful, and then finally reported back to the host.

```
USBMSC_bufferProcessed();
```

After you have explored the code, download the example to the MSP430. When it is run, the host operating system detects a CD-ROM drive and behaves identically to a physical CD-ROM drive. If autoplay information is present in the ISO file, an autoplay dialog appears when the USB Experimenter's Board is connected to the host. Figure 3 shows a flowchart of the entire read process of CDFS.

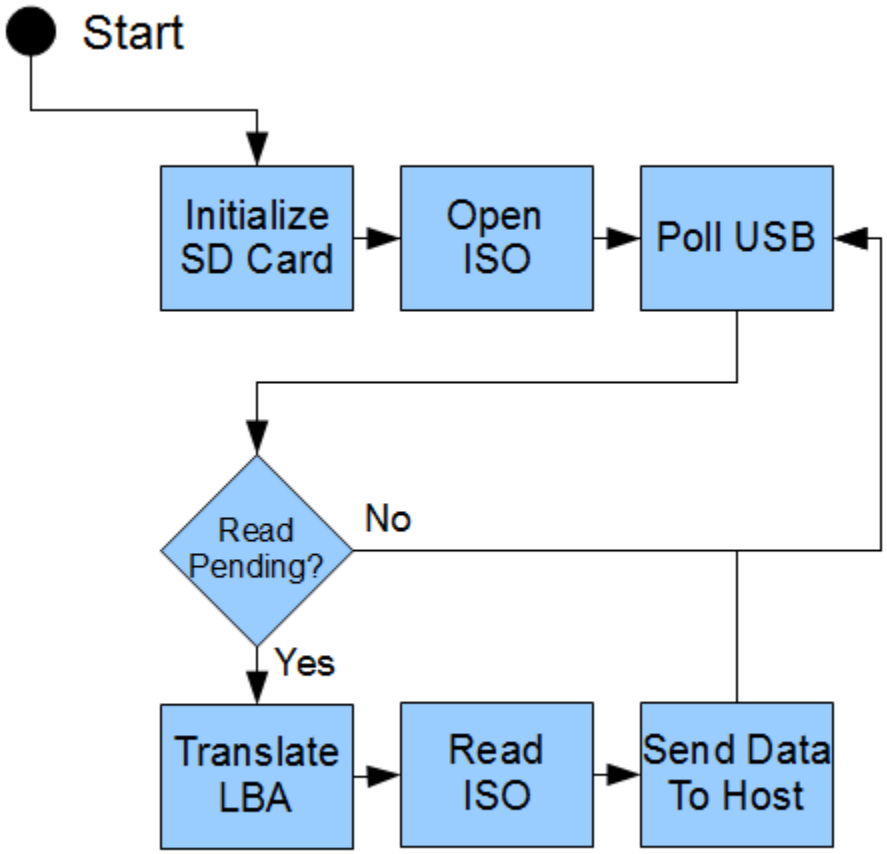

Figure 3. Flowchart of CDFS Read

3 Resources

MSP430™ Ultra-Low Power 16-Bit Microcontrollers

MSP430 Ultra-Low Power 16-bit Microcontroller Forum

MSP430 USB Developer's Package 3.11.00

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
	TI E2E Community Home Page		e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265