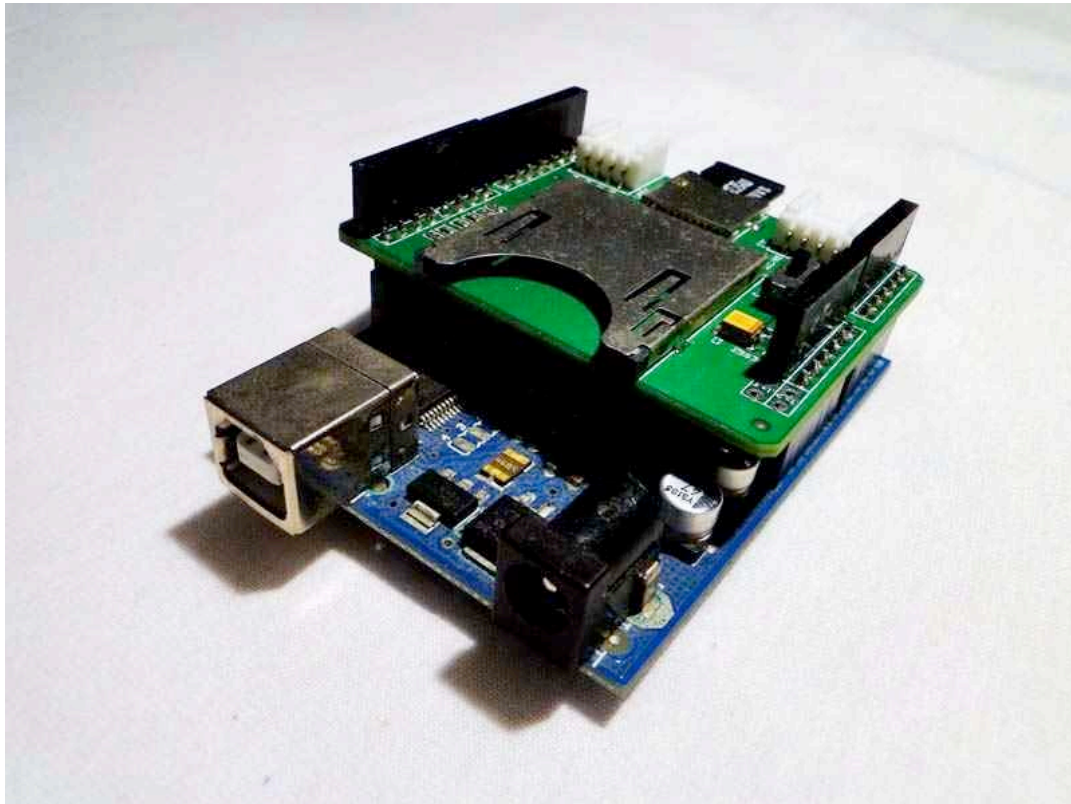


SD Card Arduino Shield Tutorial (Getting Started)



The [SD card](#) is the end of the data storage problem for the [Arduino](#), making it a must have if your project is running away from a computer. Let's say you want to make some kind of data logging in the woods, you certainly can't attach your Arduino to a computer just to store data, but with the SD card shield, you can. In this case, the Arduino will write on the SD card, no need for a computer! And all of that running off a small power supply! Neat, huh?

The SD library allows for reading from and writing to SD cards. The library supports FAT16 and FAT32 file systems on standard SD cards. It only supports one open file at a time and only uses short 8.3 file names. The file names passed to the SD library functions can include paths separated by forward-slashes, /, e.g. "directory/filename.txt". Because the working directory is always the root of the SD card, a name refers to the same file whether or not it includes a leading slash (e.g. "/file.txt" is equivalent to "file.txt").



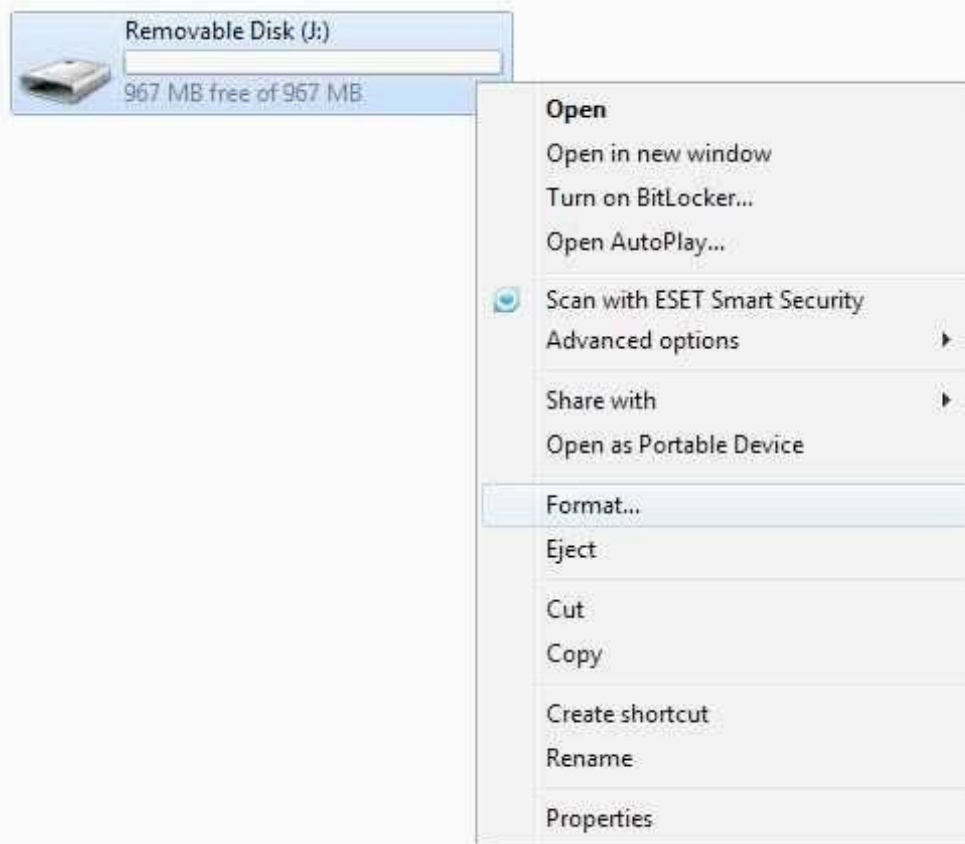
Tutorial Procedures

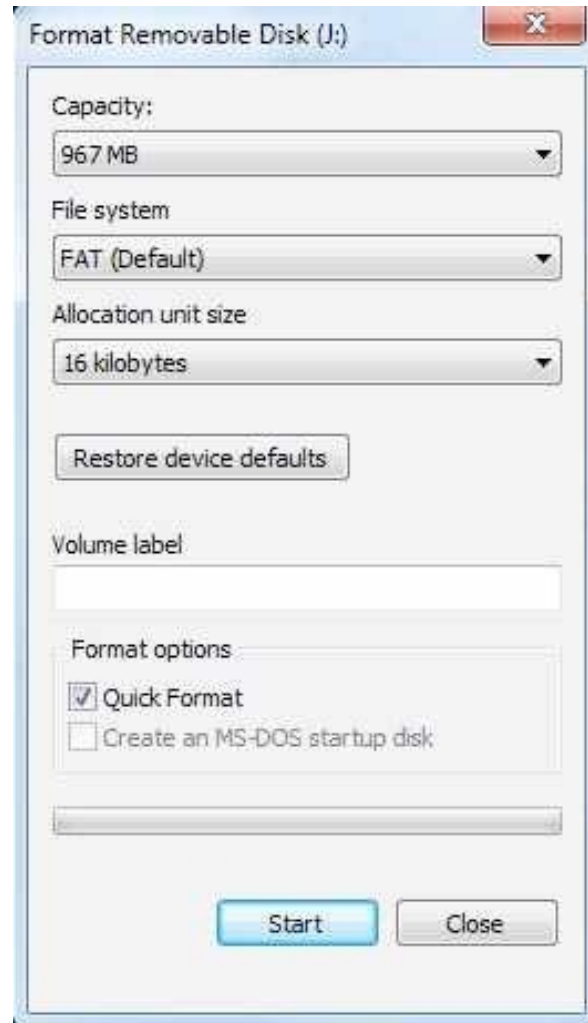
This tutorial will teach you how to get started with the shield.

The first thing you have to do is to get an SD Card. Most computers have an SD card reader available, we will use that to format the card.

The card must be formatted as FAT16. The process of formatting the card using windows is pretty straightforward, take a look at the screenshots below:

(Note: for format in Linux and Mac OS , see next section)

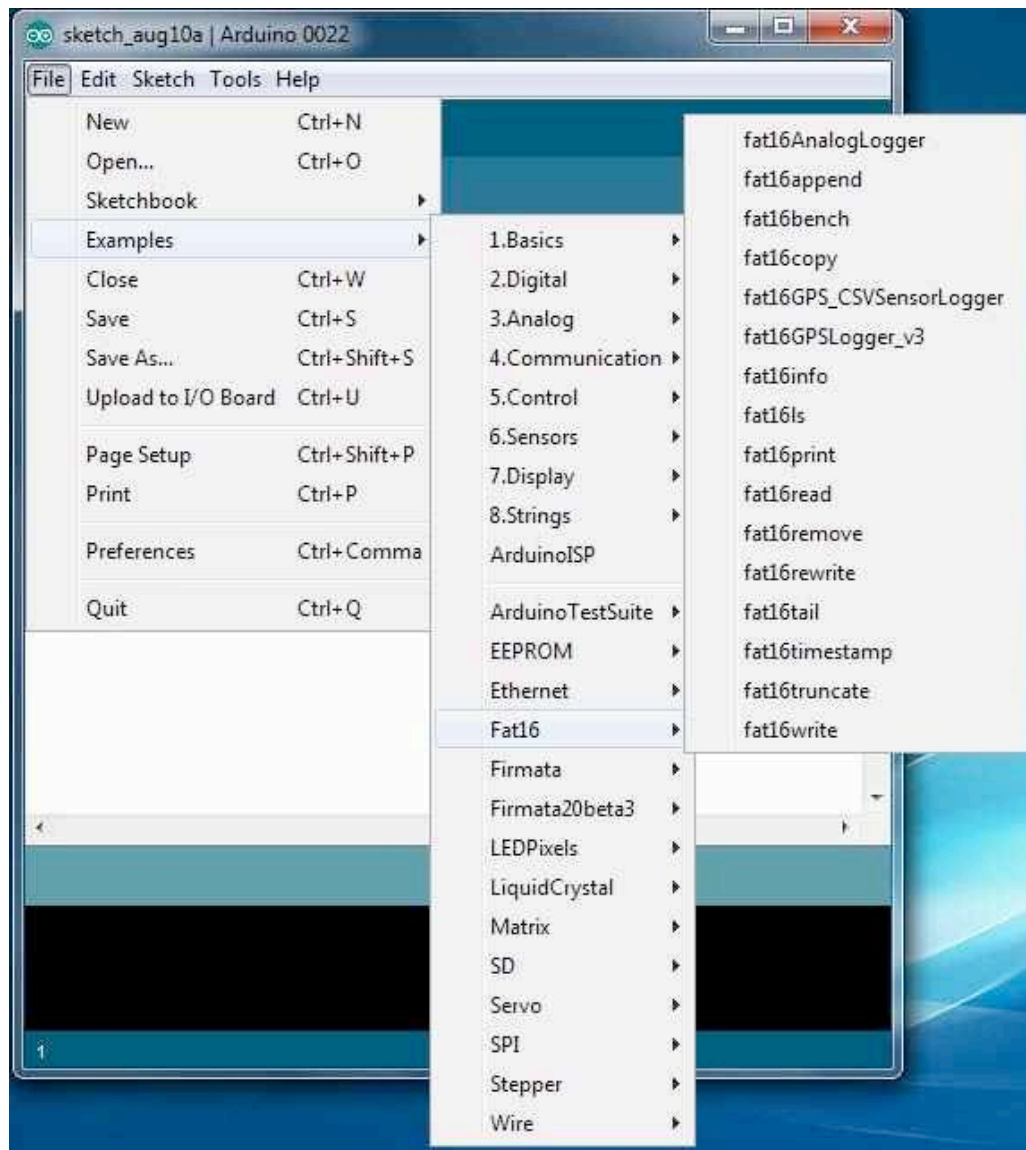




After formatting you can attach the SD card shield to your Arduino and then insert the SD card to the appropriate slot on it. Make sure the card selector is pointing the right way (Standard card or micro).

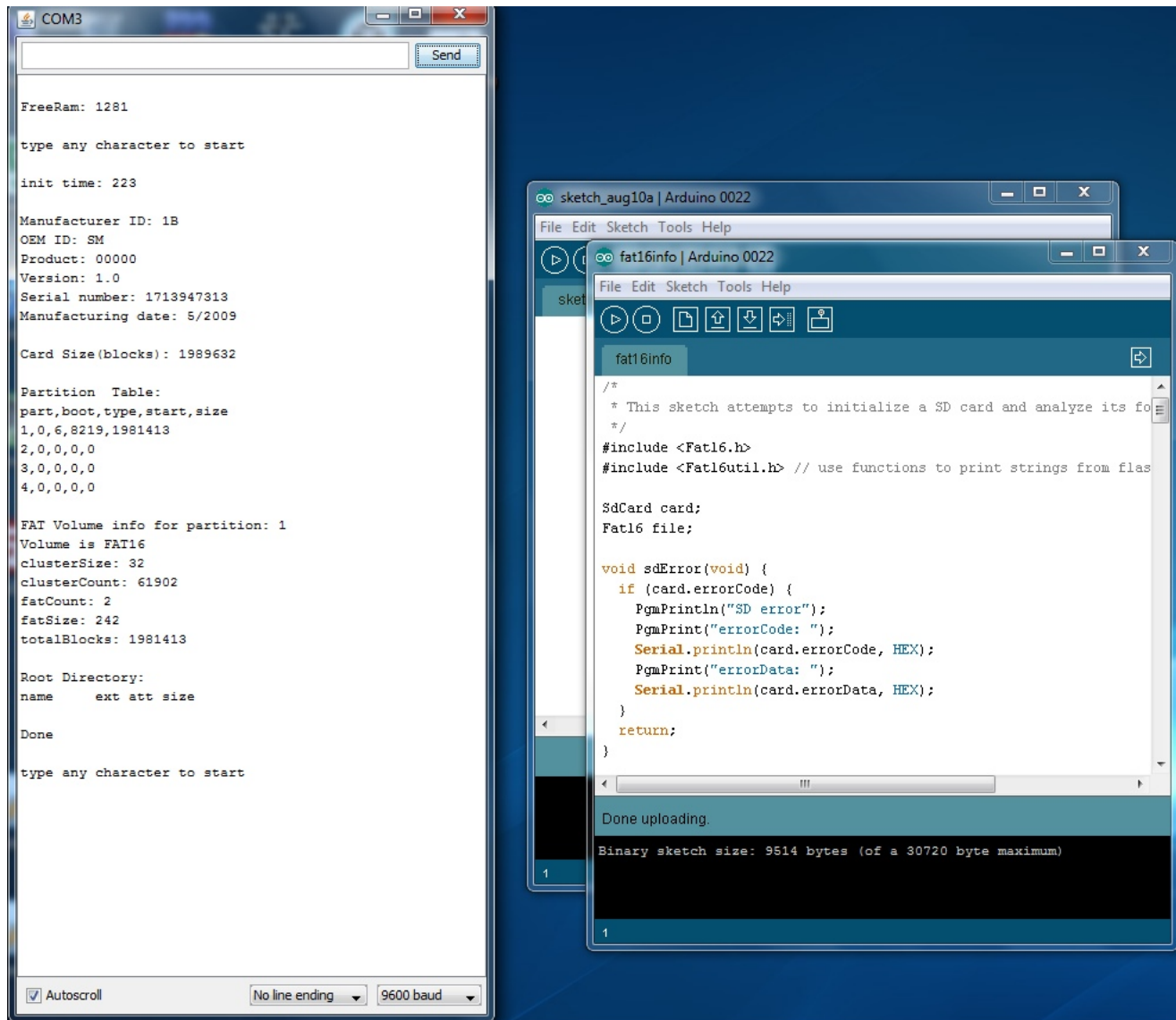
Next step is to copy the library to your Arduino library folder. After that, run Arduino IDE. Make sure the correct Arduino board and port are selected on the Tools menu.

Lets see if Arduino can see the SD card and display some info about it. For that, go to 'File', 'Examples', 'Fat16', and click 'fat16info'.



This will load a sketch that attempts to initialize the SD card and analyze its format. After uploading the sketch, open the Serial Monitor. After sending a

character the SD card information will be displayed. You should see something like this:



is working. If none of that fixes the problem, try replacing the SD card.

With that done, you can take a look at the other library examples for reading, writing, and sample logging with a GPS or real time clock. Enjoy!

Note (Formatting SD Card For different Operating systems):

OSX : Open Disk Utility (located in Applications>Utilities). Choose the Card, click on the erase tab, select MS-DOS(FAT) as the Format, and click Erase.

NB: OSX places a number of “hidden” files on the device when it formats a drive. To format a SD card without the extra files on OSX, follow these notes on Ladyada’s site.

Linux: With a SD card inserted, open a terminal window. At the prompt, type df, and press enter. The windows will report the device name of your SD card, it should look something like /dev/sdb1. Unmount the SD card, but leave it in the computer. Type sudo mkdosfs -F 16 /dev/sdb1, replacing the device name with yours. Remove the SD card and replace it to verify it works.

Note on File Naming:

FAT file systems have a limitation when it comes to naming conventions. You must use the 8.3 format, so that file names look like “NAME001.EXT”, where “NAME001” is an 8 character or fewer string, and “EXT” is a 3 character extension. People commonly use the extensions .TXT and .LOG. It is possible to have a shorter file name (for example, mydata.txt, or time.log), but you cannot use longer file names. [Read more on the 8.3 convention.](#)

Opening/Closing Files

When you use `file.write()`, it doesn't write to the card until you `flush()` or `close()`. Whenever you open a file, be sure to close it to save your data.

Make sure to close a file before opening a new one. It is only possible to have one file open at a time.

Acknowledgement:

This document was written with the help of material from Arduino Italy and Seed Studio