SD Card shield V4.0 SKU 103030005

This is the newly released SD card V4.0 shield. It provides a storage space for your Arduino. Users can read/write SD card via Arduino's built-in SD library. It supports SD, SDHC and Micro SD cards. It will only occupy the SPI port of your Arduino. Comparing to previous versions, it combines the standart SD slot and the Micro SD slot into a standard one, the included adaptor enables using of Micro SD cards . You can stack on other shields that work with the unused pins. Additionally, the preformed I2C and UART port facilitates your connection with Grove modules.

Compatibility

We have produced a lot of extension board that can make your platform board more powerful, however not every extension board is compatible with all the platform board, here we use a table to illustrate how are those boards compatible with platform board.

!!!note Please note that "Not recommended" means that it might have chance to work with the platform board however requires extra work such as jump wires or rewriting the code. If you are interested in digging more, welcome to contact with techsupport@seeed.cc.

Click to see full picture

		Arduino Uno Seeeduino v4.2	Arduino Mega Seeeduino Mega	Zero(m0) LoraWan	Arduino Leonardo Seeeduino Lite	Arduino 101	Arduino Due 3.3v	Intel Edison 5v	Linkit One
2	2.8'' TFT Touch Shield V2.0	bmp nonsupport	bmp nonsupport	Not recommended	bmp nonsupport	Not recommended	Not recommended	Not recommended	Not recommended
3	Base Shield V2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
4	Camera Shield	Only Pin234567	Hardware Serial OK	Not recommended	Not recommended	Yes	Hardware Serial OK	No	No
5	EL Shield	Yes	Yes	No	Yes	No	No	No	No
6	Energy Shield	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
7	GPRS Shield	Not recommended	Not recommended	Yes	Yes	Yes	Not recommended	Yes	No need
8	Motor Shield V2.0	Yes	Stepper motor only	No	Yes	Stepper motor only	Stepper motor only	No	No
9	Music Shield V2.0	Yes	Yes	Not recommended	Yes	Yes	Yes	Yes	Yes
10	NFC Shield V2.0	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
11	Protoshield Kit for Arduino	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
12	RS232 Shield	Yes	Yes	No	Yes	No	No	No	No
13	Relay Shield V3.0	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
14	SD Card Shield V4.0	Yes	Yes	Not recommended	Yes	Yes	Yes	No	No
15	Seeed BLE Shield V1	Yes	Not recommended	Not recommended	Yes	No need	Not recommended	Not recommended	No need
16	W5500 Ethernet Shield	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
17	Wifi Shield(Fi250) V1.1	Not recommended	Not recommended	Not recommended	Yes	Yes	Not recommended	No need	No need
18	Wifi Shield V2	Yes	Not recommended	Not recommended	Yes	Yes	Not recommended	No need	No need
19	XBee Shield V2	Yes	Not recommended	Not recommended	Yes	Yes	Not recommended	Not recommended	Not recommended

Application Ideas

If you want to make some awesome projects with SD Card Shield, here are some projects for your reference.

Here we introduce a project about LinkIt ONE using SD Card.

Music Player with LinkIt One
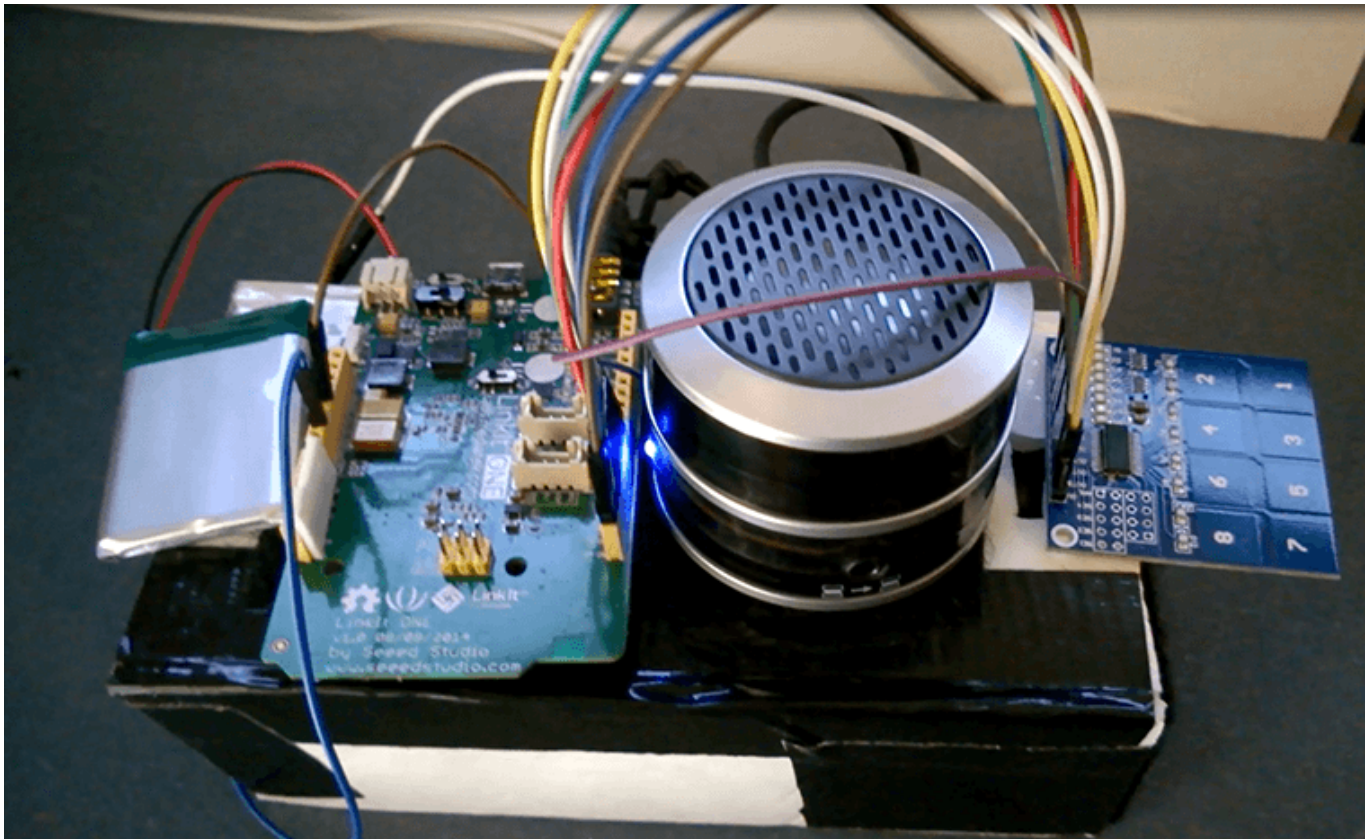
This project uses Grove - Water Sensor to create a simple but effective solution to watering plants.

Make it NOW!

More Awesome Projects by SD Card

Features

- Standard SD card, SDHC card and TF card compatible
- UART Grove & I2C connection compatible
- Fully supported SD Library
- Minimal number of SPI port
- Truly stackable

Specifications

Parameter	Value/Range
Voltage	3.5V to 5.5V
Current	0.159mA to 200mA
Supported Card Type	SD card(<=32GB); Micro SD card(<=32GB); SDHC card(<=32GB)
Dimension	68.7mmx53.5mmx19.00mm
Net Weight	14.8g
Package size	L: 102mm W: 67mm H: 31mm
Gross Weight	41g

Hardware Overview
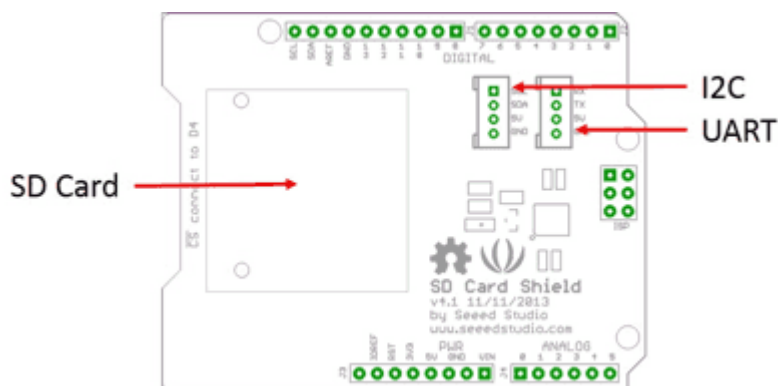
Pins usage on Arduino （**with SD card**）

D4: SD_CS;

D11: SD_DI;

D12: SD_DO;

D13: SD_CLK.

Note

The SD card format can be FAT16 or FAT32.The size of SD card and the SDHC card more than 16GB is not supported.
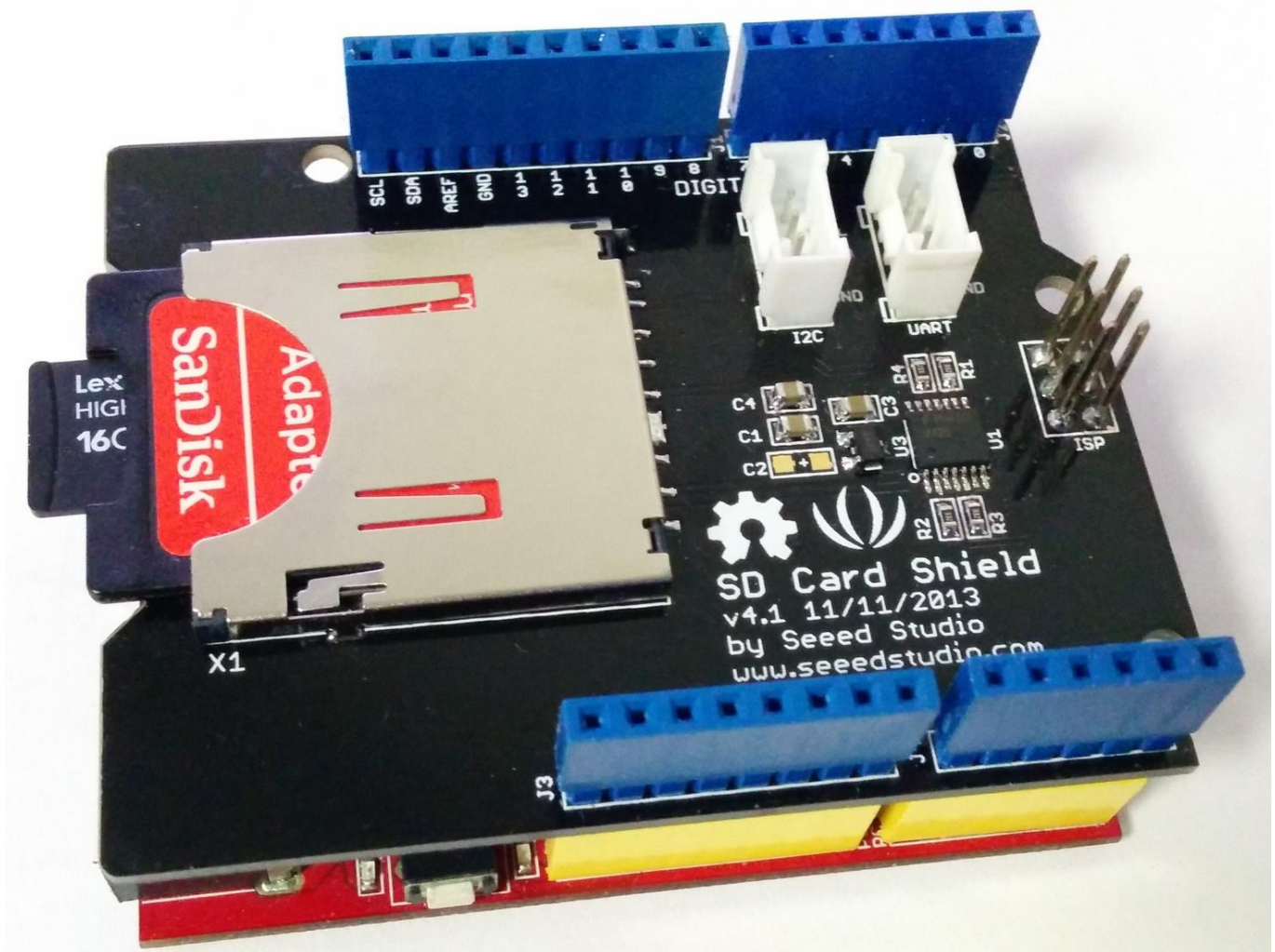
Usage

The following is the installation of the SD card shield.
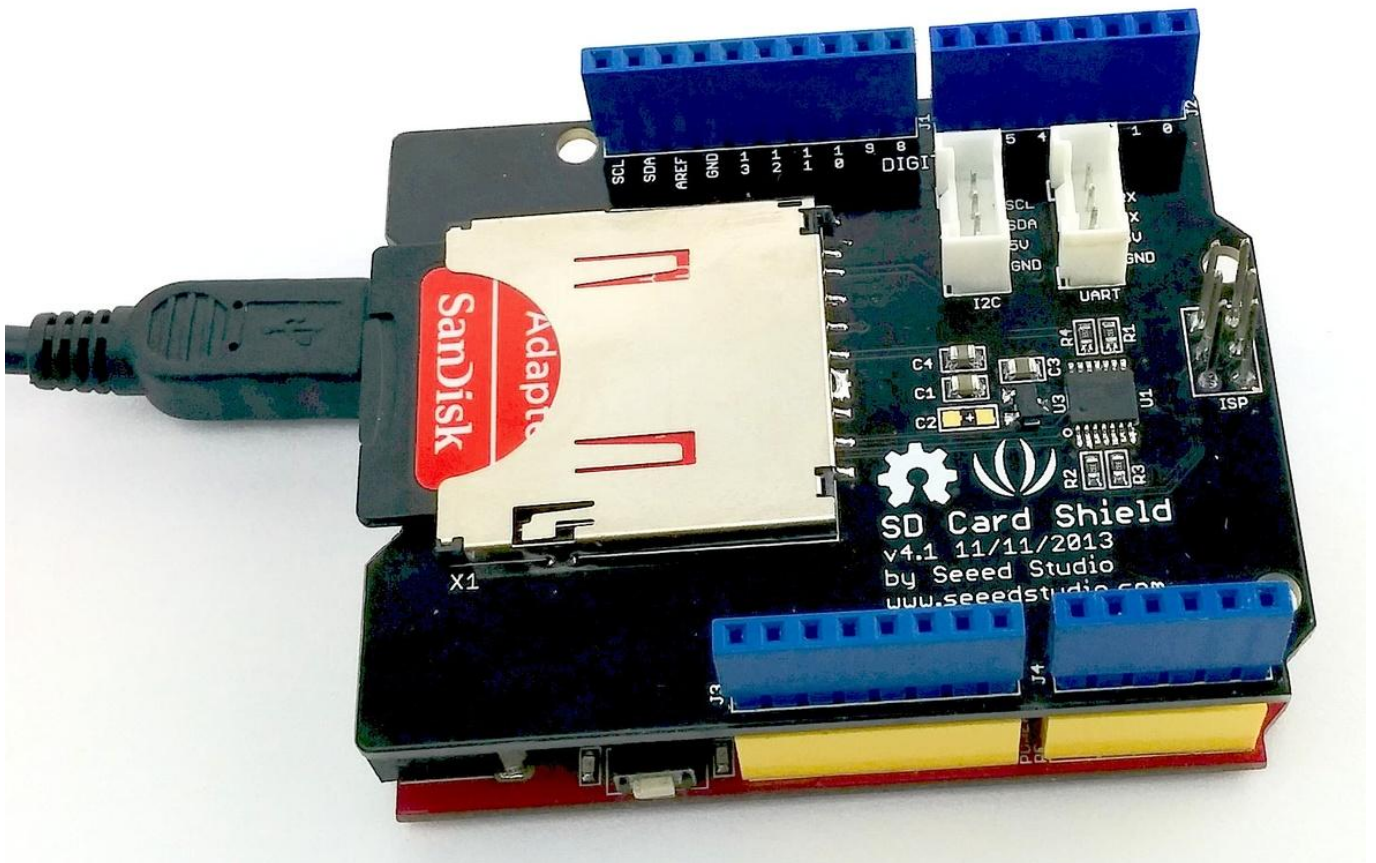
Hardware Installation

1. Insert your SD card into the socket, Plug the SD Card Shield onto the Arduino .
2. Connect the Arduino to PC using USB cable.

Note

When you use a Micro SD card, please insert Micro SD card into the adopter, and then insert the Micro SD card Adopter into the socket as shown below.

When you complete the hardware installation, it should look like this

Upload the program

1. Restart the Arduino IDE. Open "CardInfo"example via the path: File --> Examples --> SD --> CardInfo.

   This example shows how use the utility libraries on which the SD library is based in order to get info about your SD card.Very useful for testing a card when

   You are not sure whether its working or not. There are also many other examples in this library, like "ReadWrite". You can always try them out.
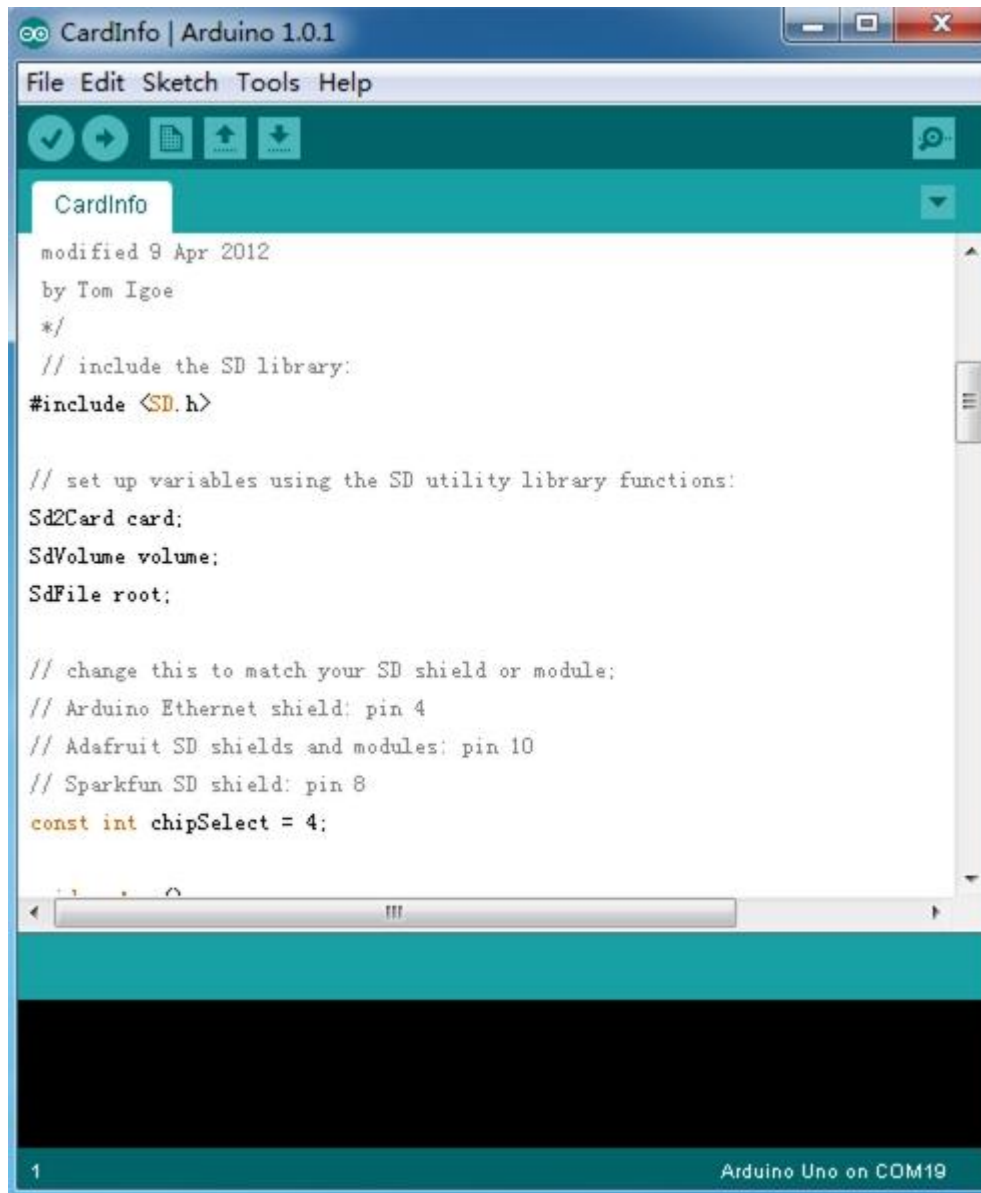
There is a brief description to above code:

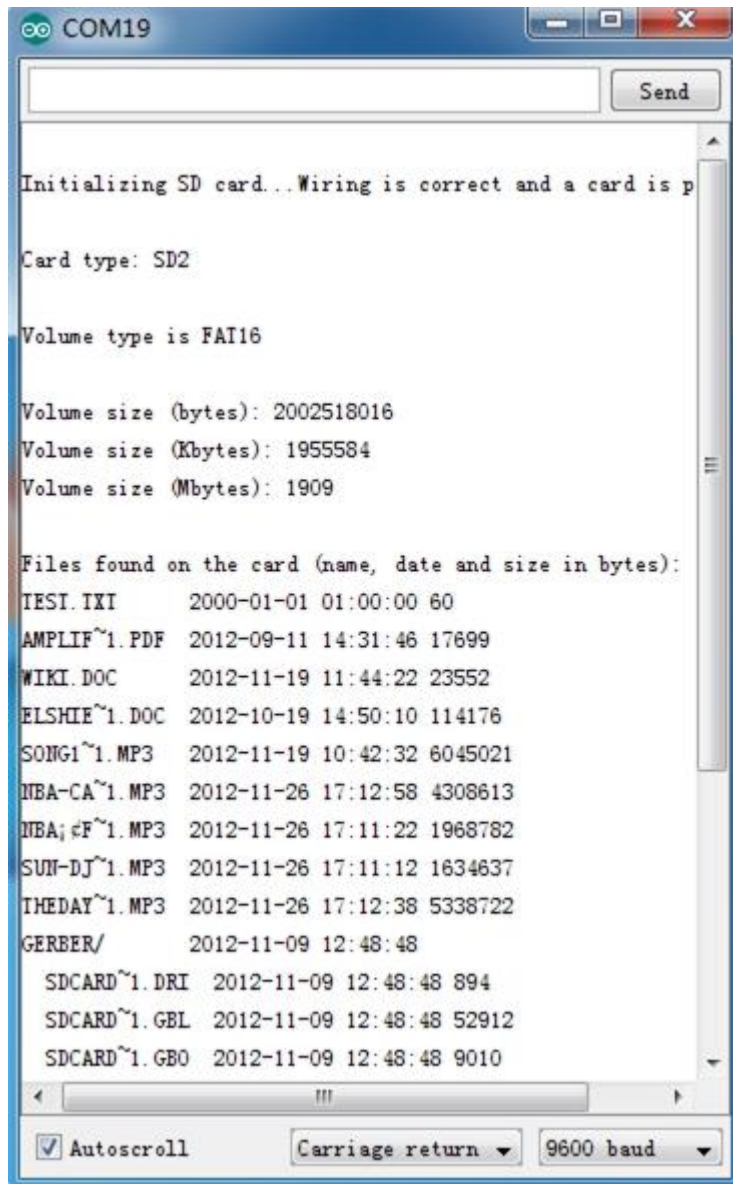First check if the card is working. If not, there will output some reasons that may lead to this result.

In the case of the SD card normal work, it will print the SD card type.Then print the type and size of the FAT-type volume.

Finally, get the files information such as name, date and size in bytes that found on the card.

2. Upload the code.

3. View Results. You can see the follow image after Click the serial monitor.

```
COM19                                          —  □  X

┌──────────────────────────────────┐ ┌──────────┐
│                                  │ │   Send   │
└──────────────────────────────────┘ └──────────┘

Initializing SD card...Wiring is correct and a card is p

Card type: SD2

Volume type is FAT16

Volume size (bytes): 2002518016
Volume size (Kbytes): 1955584
Volume size (Mbytes): 1909

Files found on the card (name, date and size in bytes):
TEST.TXT       2000-01-01 01:00:00 60
AMPLIF~1.PDF  2012-09-11 14:31:46 17699
WIKI.DOC       2012-11-19 11:44:22 23552
ELSHIE~1.DOC  2012-10-19 14:50:10 114176
SONG1~1.MP3   2012-11-19 10:42:32 6045021
NBA-CA~1.MP3  2012-11-26 17:12:58 4308613
NBA;¢F~1.MP3  2012-11-26 17:11:22 1968782
SUN-DJ~1.MP3  2012-11-26 17:11:12 1634637
THEDAY~1.MP3  2012-11-26 17:12:38 5338722
GERBER/        2012-11-09 12:48:48
  SDCARD~1.DRI  2012-11-09 12:48:48 894
  SDCARD~1.GBL  2012-11-09 12:48:48 52912
  SDCARD~1.GB0  2012-11-09 12:48:48 9010

◄              III                    ►

☑ Autoscroll      Carriage return ▼   9600 baud  ▼
```

4. If an error occurs, please recheck all the previous steps, and make sure the SD card is working. If none of that fixes the problem, try replacing the SD card.

Note

Arduino default code returns SD size incorrectly if your SD is card more than 4G. The following code can solve this issue.

```
/*
  SD card test

  This example shows how use the utility libraries on which the'
  SD library is based in order to get info about your SD card.
  Very useful for testing a card when you're not sure whether its working or
not.

  The circuit:
   * SD card attached to SPI bus as follows:
  ** MOSI - pin 11 on Arduino Uno/Duemilanove/Diecimila
  ** MISO - pin 12 on Arduino Uno/Duemilanove/Diecimila
```

```
    ** CLK - pin 13 on Arduino Uno/Duemilanove/Diecimila
    ** CS - depends on your SD card shield or module.
    ** Pin 4 used here for consistency with other Arduino examples


  created  28 Mar 2011
  by Limor Fried
  modified 9 Apr 2012
  by Tom Igoe
  */
// include the SD library:
#include <SPI.h>
#include <SD.h>

// set up variables using the SD utility library functions:
Sd2Card card;
SdVolume volume;
SdFile root;

// change this to match your SD shield or module;
// Arduino Ethernet shield: pin 4
// Adafruit SD shields and modules: pin 10
// Sparkfun SD shield: pin 8
const int chipSelect = 4;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }


  Serial.print("\nInitializing SD card...");
  // On the Ethernet Shield, CS is pin 4. It's set as an output by default.
  // Note that even if it's not used as the CS pin, the hardware SS pin
  // (10 on most Arduino boards, 53 on the Mega) must be left as an output
  // or the SD library functions will not work.
  pinMode(10, OUTPUT);     // change this to 53 on a mega


  // we'll use the initialization code from the utility libraries
  // since we're just testing if the card is working!
  if (!card.init(SPI_HALF_SPEED, chipSelect)) {
    Serial.println("initialization failed. Things to check:");
    Serial.println("* is a card is inserted?");
    Serial.println("* Is your wiring correct?");
    Serial.println("* did you change the chipSelect pin to match your shield
or module?");
    return;
  } else {
    Serial.println("Wiring is correct and a card is present.");
  }
```

```
      // print the type of card
      Serial.print("\nCard type: ");
      switch (card.type()) {
        case SD_CARD_TYPE_SD1:
          Serial.println("SD1");
          break;
        case SD_CARD_TYPE_SD2:
          Serial.println("SD2");
          break;
        case SD_CARD_TYPE_SDHC:
          Serial.println("SDHC");
          break;
        default:
          Serial.println("Unknown");
      }

      // Now we will try to open the 'volume'/'partition' - it should be FAT16 or
FAT32
      if (!volume.init(card)) {
        Serial.println("Could not find FAT16/FAT32 partition.\nMake sure you've
formatted the card");
        return;
      }


      // print the type and size of the first FAT-type volume

      uint64_t volumesize64;
      uint32_t volumesize32;
      Serial.print("\nVolume type is FAT");
      Serial.println(volume.fatType(), DEC);
      Serial.println();

      volumesize64 = volume.blocksPerCluster();    // clusters are collections of
blocks
      volumesize64 *= volume.clusterCount();        // we'll have a lot of clusters
      volumesize64 *= 512;                           // SD card blocks are always
512 bytes

      Serial.print("Volume size (bytes): ");
      printLLNumber(volumesize64, DEC);
      Serial.println();

      Serial.print("Volume size (Kbytes): ");
      volumesize32 = volumesize64/1024;
      Serial.println(volumesize32);

      Serial.print("Volume size (Mbytes): ");
      volumesize32 /= 1024;
      Serial.println(volumesize32);
      /*uint64_t volumesize;
      Serial.print("\nVolume type is FAT");
      Serial.println(volume.fatType(), DEC);
```

```
      Serial.println();

      volumesize = volume.blocksPerCluster();    // clusters are collections of
  blocks
      volumesize *= volume.clusterCount();        // we'll have a lot of clusters
      volumesize *= 512;                            // SD card blocks are always
  512 bytes
      Serial.print("Volume size (bytes): ");
      Serial.println(volumesize,DEC);
      Serial.print("Volume size (Kbytes): ");
      volumesize /= 1024;
      Serial.println(volumesize,DEC);
      Serial.print("Volume size (Mbytes): ");
      volumesize /= 1024;
      Serial.println(volumesize,DEC);
    */

      Serial.println("\nFiles found on the card (name, date and size in bytes):
  ");
      root.openRoot(volume);

      // list all files in the card with date and size
      root.ls(LS_R | LS_DATE | LS_SIZE);
    }


    void loop(void) {

    }
    void printLLNumber(uint64_t n, uint8_t base)
    {
      unsigned char buf[16 * sizeof(long)];
      unsigned int i = 0;

      if (n == 0)
      {
        Serial.print((char)'0');
        return;
      }

      while (n > 0)
      {
        buf[i++] = n % base;
        n /= base;
      }

      for (; i > 0; i--)
        Serial.print((char) (buf[i - 1] < 10 ?
          '0' + buf[i - 1] :
          'A' + buf[i - 1] - 10));
    }
```

Resources

- [SD Card Shield v4.0 Schematic](#)

- [SD Card Shield v4.0 Eagle File.zip](#)

- [SD Card Shield v4.0a Eagle File.zip](#)

- [SD Card Shield v4.3 Eagle file.zip](#)

Tech Support

Please submit any technical issue into our [forum](#) or drop mail to techsupport@seeed.cc.