

Arduino Driver SD Card

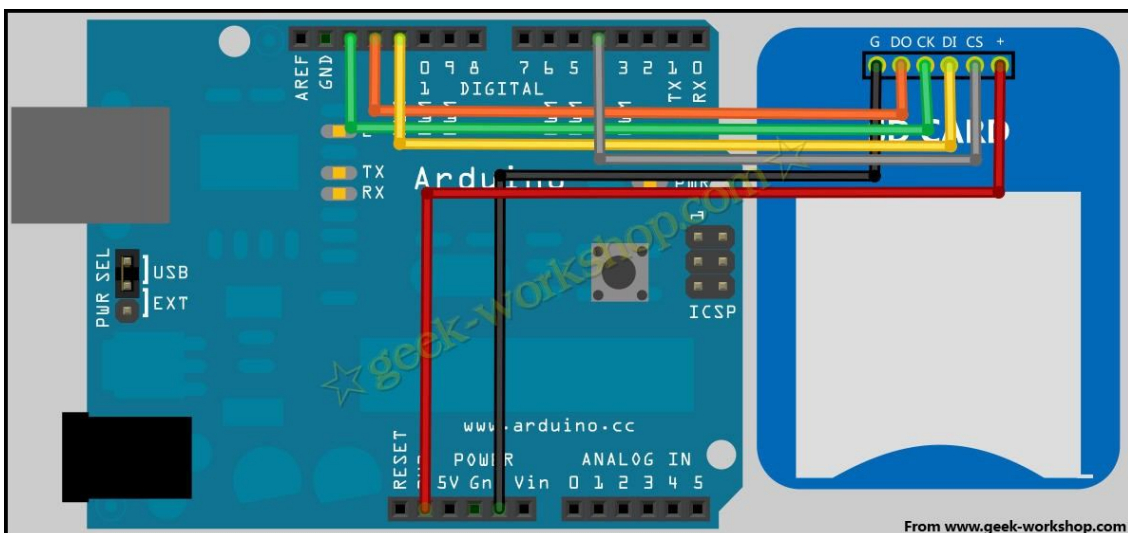
The arduino driver SD card is for the file's reading and writing

What requires special explanation is the SD library file. Currently, it can't support the card over 2G well, so I recommend 2G or less than 2G while the file is FAT format.

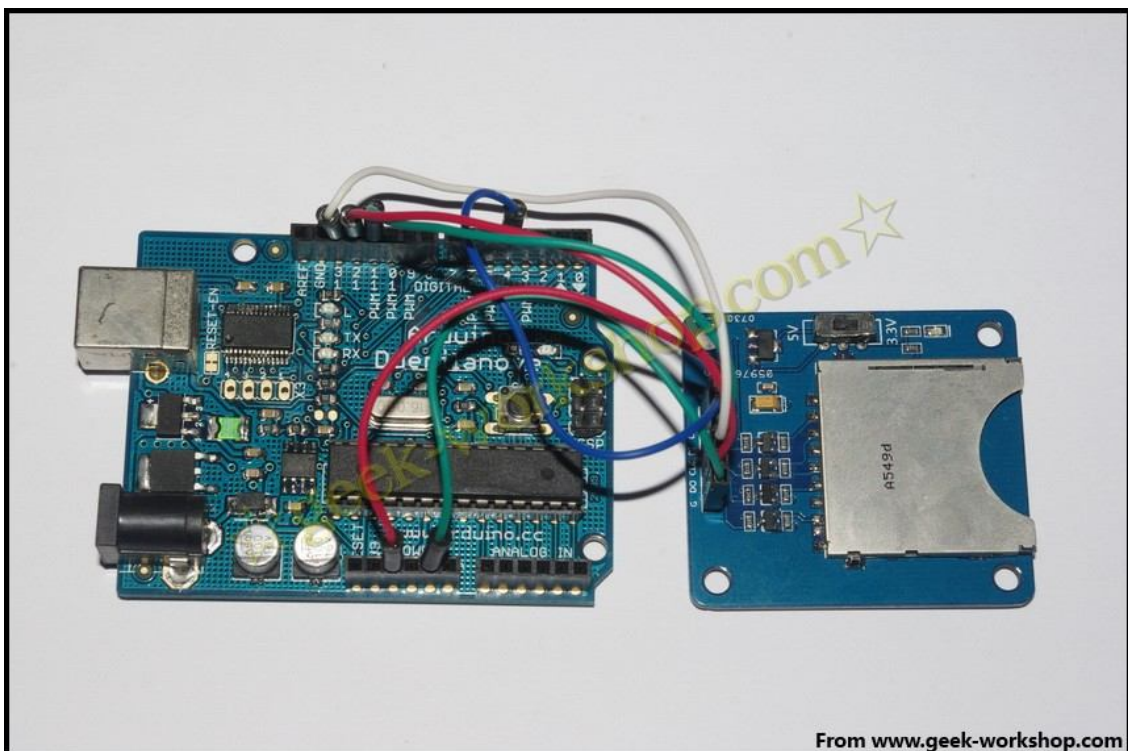
Therefore, I use the MicroSD card of kingmax2G.



The connection diagram of hardware:



2011-9-10 13:38



ARDUINO SKETCH 1

/*

Record 3 stimulated pins to the SD card and use SD library file.
The stimulated pins connect the stimulated port 0, 1, 2. Connect to the stimulated sensor depends on your selection. However, if the above steps are omitted, the experiment can still be completed but the unceasingly jumping data will bother you.

SD card and arduino controlling board connect to the main SPI wire

** MOSI - pin 11

** MISO - pin 12

** CLK - pin 13

** CS - pin 4

*/

```
#include <SD.h>
```

```
// The CS pin is pin4, and the official default settings are also used here.
```

```
const int chipSelect = 4; // Set CS interface
```

```
// The pin of the CS is pin4. A default setting here.
```

```
const int chipSelect = 4; //Set it as CS port
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600); // Set the serial communication baud rate 9600
```

```
  Serial port baud rate: 9600
```

```
  Serial.print("Initializing SD card..."); // Serial port output date: Initializing SD card...
```

```
  pinMode(10, OUTPUT);
```

```
  if (!SD.begin(chipSelect)) { //If the communication from the CS port and the SD card fails,  
the serial port outputs information.
```

```
Card failed, or not present
```

```
{ //If the communication fails from the CS port to the SD card, the serial port will output the  
message: Card failed, or not present
```

```
  Serial.println("Card failed, or not present");
```

```
  return;
```

```
}
```

```
  Serial.println("card initialized."); //Successful communication with SD card, serial output  
information
```

```
card initialized.
```

```
}
```

```
//If the communication succeeds , the serial port will output the message: card initialized.
```

```
}
```

```
void loop()
```

```
{
```

```
  // Define array
```

```
  String dataString = "";
```

```
  // Read the data of 3 sensors and write down the data.
```

```
  for (int analogPin = 0; analogPin < 3; analogPin++) {
```

```
    int sensor = analogRead(analogPin);
```

```
    dataString += String(sensor);
```

```
    if (analogPin < 2) {
```

```
      dataString += ", ";
```

```

    }
}

// Open the file. Notice! You can only open one file at a time.
//File dataFile = SD.open("datalog.txt", FILE_WRITE);

// Open datalog.txt , reading and writing status , The file is located at the bottom of the file.
if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    // Array dataString output to serial port
    Serial.println(dataString);
}

// If you can't open the file , the serial port will output the error message: error opening
datalog.txt
else {
    Serial.println("error opening datalog.txt");
}
}

```

ARDUINO SKETCH 2

```

/*
This example shows that how to record 3 stimulated pins to the SD card and use the SD
library file
The stimulated pins connect the stimulated port 0, 1, 2. Connect to the stimulated sensor
depends on your selection. However, if the above steps are omitted, the experiment can still
be completed but the unceasingly jumping data will bother you.
SD card and arduino controlling board connect to the main SPI wire
** MOSI - pin 11
** MISO - pin 12
** CLK - pin 13
** CS - pin 4
*/

#include <SD.h>

// the CS pin is pin4, and the official default settings are also used here.
const int chipSelect = 4; // Set CS interface

// The pin of the CS is pin4. A default setting here.
const int chipSelect = 4; //Set it as CS port

void setup()
{
    Serial.begin(9600); //Set the serial communication baud rate baud rate9600
    Serial.print("Initializing SD card..."); // the serial port will output the data: Initializing SD card...
    pinMode(10, OUTPUT);
    if (!SD.begin(chipSelect)) { //If the communication from the CS port and the SD card fails, the
serial port outputs information. Card failed, or not present If the communication fails from the
CS port to the SD card, the serial port will output the message: Card failed, or not present

```

```

Serial.println("Card failed, or not present");
return;
}
Serial.println("card initialized."); //Successful communication with SD card, serial output
information card initialized.
}

void loop()
{
// Define array
String dataString = "";
//Read the data of 3 sensors and write down the data.
for (int analogPin = 0; analogPin < 3; analogPin++) {
int sensor = analogRead(analogPin);
dataString += String(sensor);
if (analogPin < 2) {
dataString += ",";
}
}
} //END

```

//Open the file. Notice! You can only open one file at a time.
// If you want to open another file, you need to close the previous one.

```

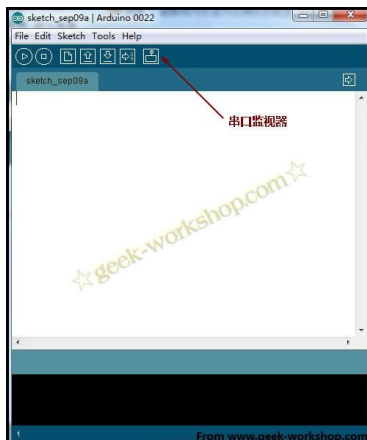
File dataFile = SD.open("datalog.txt", FILE_WRITE);

// Open datalog.txt , reading and writing status , The file is located at the end of the file.

if (dataFile) {
dataFile.println(dataString);
dataFile.close();
// Array dataString output to the serial port
Serial.println(dataString);
}
// If you can't open the file, the serial port will output error opening datalog.txt
else {
Serial.println("error opening datalog.txt");
}
}

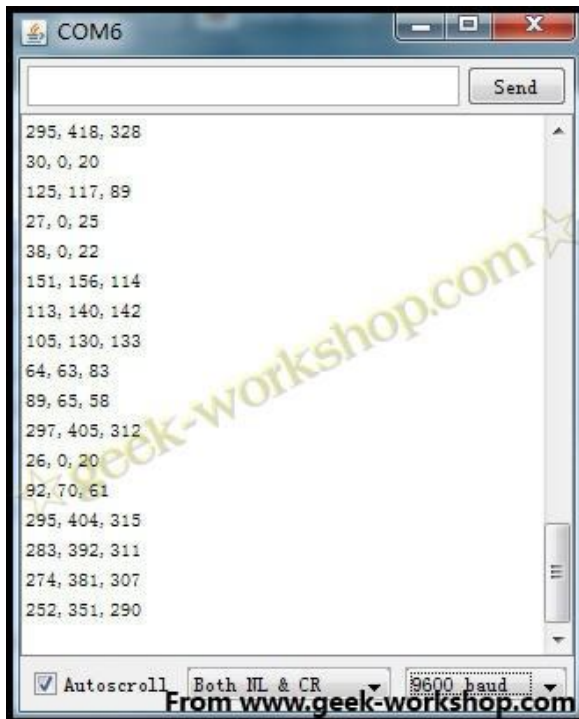
```

Open the serial monitor.



After the control panel starts working, you will see the following screen. Each row in the figure is the three sensor values recorded each time. This experiment does not connect the sensor, so the value is the messy signal of the surrounding interference source.

The control panel starts working.



Switch off the arduino, take out the SD card and connect to the computer. There is a DATALOG.TXT in the SD card. You could check the data of each sensor.



After the experiment of the first code, don't delete the DATALOG.TXT file in SD card, and download the following code to the arduino.

ARDUINO SKETCH 3

/*

This example shows that how to read a file from the SD card through SD library file and send the data through serial port.

* SD card and arduino control board are connected using SPI bus

** MOSI - pin 11

** MISO - pin 12

```

** CLK - pin 13
** CS - pin 4
*/

#include <SD.h>

// CS pin is pin4 , The official default settings are also used here.

const int chipSelect = 4; // Set CS interface

void setup()
{
  Serial.begin(9600); // Set the serial communication baud rate 9600
  Serial.print("Initializing SD card..."); // Serial output data Initializing SD card...
  pinMode(10, OUTPUT);

  if (!SD.begin(chipSelect)) { // If the communication between the CS port and the SD card fails, the
serial port output information Card failed, or not present
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  Serial.println("card initialized."); // Successful communication with SD card, serial output
information card initialized.

  File dataFile = SD.open("datalog.txt"); // Open the datalog.txt file
  if (dataFile) {
    while (dataFile.available()) { // Check if dataFile has data
      Serial.write(dataFile.read()); // Send data to the serial port if there is data
    }
    dataFile.close(); // Close dataFile
  }
  else {
    Serial.println("error opening datalog.txt"); // If the file cannot open the serial port, send
information error opening datalog.txt
  }
}

void loop()
{
}

```

ARDUINO SKETCH 4

```

/*
This example shows that how to read a file from the SD card through SD library file and send
the data through serial port.
* SD card and arduino control board are connected using SPI bus
SD card and arduino controlling board connect to the main SPI wire
** MOSI - pin 11
** MISO - pin 12
** CLK - pin 13
** CS - pin 4
*/

```

```

#include <SD.h>
// The CS pin is pin4, and the official default settings are also used here.
const int chipSelect = 4; // Set CS interface

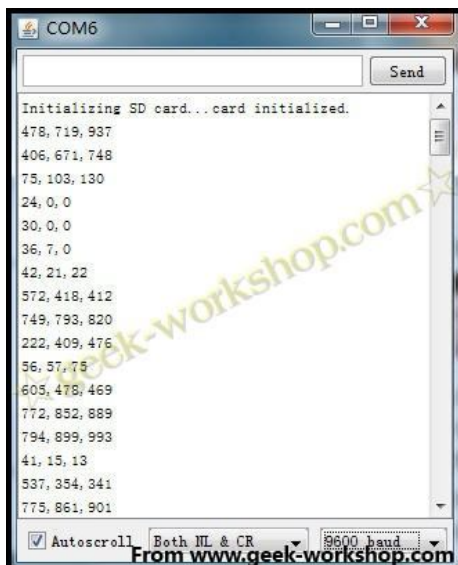
void setup()
{
  Serial.begin(9600); // Set the serial communication baud rate to 9600
  Serial.print("Initializing SD card..."); // Serial output data Initializing SD card...
  pinMode(10, OUTPUT);
  if (!SD.begin(chipSelect)) { // If the communication between the CS port and the SD card fails, the
serial port output information Card failed, or not present
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  Serial.println("card initialized."); // Successful communication with the SD card, serial output
information card initialized.
  File dataFile = SD.open("datalog.txt"); // Open the datalog.txt file
  if (dataFile) {
    while (dataFile.available()) { //Check the dataFile to see whether there is data.
      Serial.write(dataFile.read()); //Send the data to the serial port if data exists.
    }

    dataFile.close(); //Turn off dataFile
  }
  else {
    Serial.println("error opening datalog.txt"); // If the file cannot open the serial port, send
information error opening datalog.txt
  }
}

void loop()
{
}

```

The data of the DATALOG.TXT is sent to the computer through serial port.



Type the code to the arduino control panel and open serial monitor.

ARDUINO SKETCH 5

```
/*
This example shows that create a delete file in SD card,
* SD card and arduino controlling board connect to the main SPI wire
** MOSI - pin 11
** MISO - pin 12
** CLK - pin 13
** CS - pin 4
*/
#include <SD.h>

File myFile;

void setup()
{
  Serial.begin(9600); // Set the serial communication baud rate9600
  Serial.print("Initializing SD card..."); // Serial output data Initializing SD card...
  pinMode(10, OUTPUT);

  if (!SD.begin(4)) { // If the communication from the CS port fails with the SD card, the serial port
output information initialization failed!
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done."); // The CS port communicates successfully with the SD card, and
the serial port outputs the information initialization done.
  if (SD.exists("example.txt")) { // Check if the example.txt file exists Check the file whether exists or
not
    Serial.println("example.txt exists."); //If exists, send example.txt exists. to serial port
  }
  else {
    Serial.println("example.txt doesn't exist."); // If do not exist, send example.txt example.txt
doesn't exist to serial port  }

  Serial.println("Creating example.txt...");
  myFile = SD.open("example.txt", FILE_WRITE); //Open example.txt, If not, create one and
enter into writing mode. Write status if not automatically created
  myFile.close(); // Close off file.
  if (SD.exists("example.txt")) {
    Serial.println("example.txt exists.");
  }
  else {
    Serial.println("example.txt doesn't exist.");
  }
  Serial.println("Removing example.txt...");
  SD.remove("example.txt"); //delete example.txt
  if (SD.exists("example.txt")){
    Serial.println("example.txt exists.");
  }
  else {
    Serial.println("example.txt doesn't exist.");
  }
}
```



```
}  
}
```

```
void loop()  
{  
  
}
```

ARDUINO SKETCH 6

```
/*  
This example shows how to create a deleted file in SD card.  
* SD card and arduino controlling board connect to the main SPI wire  
** MOSI - pin 11  
** MISO - pin 12  
** CLK - pin 13  
** CS - pin 4  
*/  
#include <SD.h>  
File myFile;  
  
void setup()  
{  
  Serial.begin(9600); // Set the serial communication baud rate  
  9600  
  Serial.print("Initializing SD card..."); // Serial output data Initializing SD card...  
  pinMode(10, OUTPUT);  
  if (!SD.begin(4)) { // If the communication from the CS port fails with the SD card, the serial port  
output information initialization failed!  
    Serial.println("initialization failed!");  
    return;  
  }  
  Serial.println("initialization done."); // he CS port communicates successfully with the SD card, and  
the serial port outputs the information initialization done.  
  if (SD.exists("example.txt")) { // Check if the example.txt file exists  
    Serial.println("example.txt exists."); // f there is output information example.txt exists. to the  
serial port  
  }  
  else {  
    Serial.println("example.txt doesn't exist."); // There is no output information example.txt doesn't  
exist. to the serial port  
  }  
  Serial.println("Creating example.txt...");  
  myFile = SD.open("example.txt", FILE_WRITE); // Open example.txt, if not created automatically,  
write status  
  myFile.close(); // Close file  
  if (SD.exists("example.txt")) {  
    Serial.println("example.txt exists.");  
  }  
  else {  
    Serial.println("example.txt doesn't exist.");  
  }  
}
```

```

Serial.println("Removing example.txt...");
SD.remove("example.txt"); // Delete file example.txt
if (SD.exists("example.txt")){
    Serial.println("example.txt exists.");
}
else {
    Serial.println("example.txt doesn't exist.");
}
}

void loop() {

}

```

The figure below shows the result of the serial monitor display, showing the process of creating a file in the delete process.

The picture below is the result of the serial monitor:



)

Then pass the following code to the arduino control board to see the effect.

ARDUINO SKETCH 7

```

#include <SD.h>
File myFile;
void setup()
{
    Serial.begin(9600);
    Serial.print("Initializing SD card...");
    pinMode(10, OUTPUT);

```

```

if (!SD.begin(4)) {
  Serial.println("initialization failed!");
  return;
}
Serial.println("initialization done.");
myFile = SD.open("test.txt", FILE_WRITE);
if (myFile) {
  Serial.print("Writing to test.txt...");
  myFile.println("testing 1, 2, 3.");
  myFile.close();
  Serial.println("done.");
} else {
  Serial.println("error opening test.txt");
}
myFile = SD.open("test.txt");
if (myFile) {
  Serial.println("test.txt:");
  while (myFile.available()) {
    Serial.write(myFile.read());
  }
  myFile.close();
} else {
  Serial.println("error opening test.txt");
}
}

void loop()
{

}

#include <SD.h>
File myFile;

void setup()

{

  Serial.begin(9600);

  Serial.print("Initializing SD card...");
  pinMode(10, OUTPUT);
  if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  myFile = SD.open("test.txt", FILE_WRITE);
  if (myFile) {
    Serial.print("Writing to test.txt...");

```

```

myFile.println("testing 1, 2, 3.");
myFile.close();
Serial.println("done.");
} else {
    Serial.println("error opening test.txt");
}
myFile = SD.open("test.txt");
if (myFile) {
    Serial.println("test.txt:");
    while (myFile.available()) {
        Serial.write(myFile.read());
    }
    myFile.close();
} else {
    Serial.println("error opening test.txt");
}
}
}

```

```
void loop()
```

```
{
}
```

The following screen can be seen in the serial monitor.

The picture below is the result of the serial monitor:



2011-9-10 13:38

Connect the SD card to the computer, you can find the TEST.TXT file inside, the contents are as follows.

Connect the SD card to the computer and you will find TEST.TXT:

