SYNACORP TRADING & SERVICES
No.9, 1st Floor, Lorong 1/SS2, Bandar Tasek Mutiara, 14120 Simpang Ampat, S.Prai (S), Penang
Tel : +604.502.1726 Hunting Line : 012.403.3474 Fax : +604.502.1726
(Website) http://www.synacorp.my (Email) sales@synacorp.com.my

Micro SD Card Adapter Reader Module

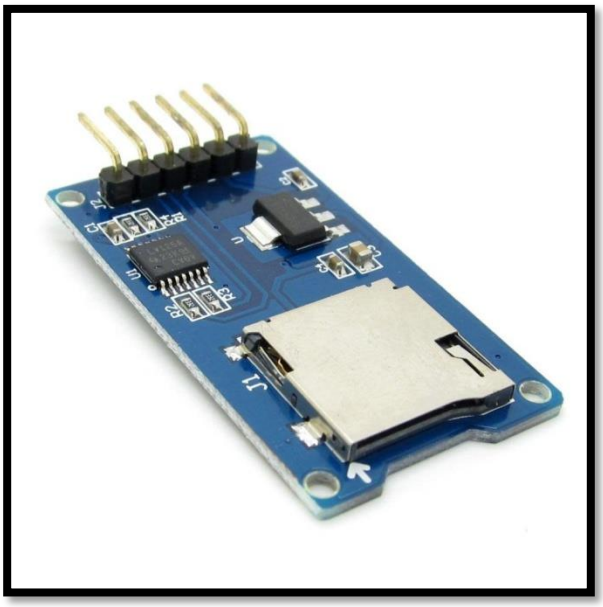
Introduction

The module (MicroSD Card Adapter) is a Micro SD card reader module, and the SPI interface via the file system driver, microcontroller system to complete the MicroSD card read and write files. Arduino users can directly use the Arduino IDE comes with an SD card to complete the library card initialization and read-write

Specification

- Supports micro SD card (<=2G), micro SDHC card (<=32G) (high-speed card)
- Level conversion circuit board that can interface level is 5V or 3.3V
- Power supply is 4.5V ~ 5.5V, 3.3V voltage regulator circuit board
- Communication interface is a standard SPI interface
- 4 M2 screw positioning holes for easy installation
- Size: 4.1 x 2.4cm
- A total of six pins (GND, VCC, MISO, MOSI, SCK, CS), GND to ground, VCC is the power supply, MISO, MOSI, SCK is the SPI bus, CS is the chip select signal pin.
- LDO regulator output 3.3V as level converter chip, Micro SD card supply.
- Micro SD card into the direction of signals into 3.3V, MicroSD card toward the direction of the control interface MISO signal is also converted to 3.3V, general AVR microcontroller system can read the signal
- 4 M2 screws positioning hole diameter of 2.2mm, the module is easy to install positioning, to achieve inter-module combination

Images



Packing List

- Micro SD Card Adapter Reader Module

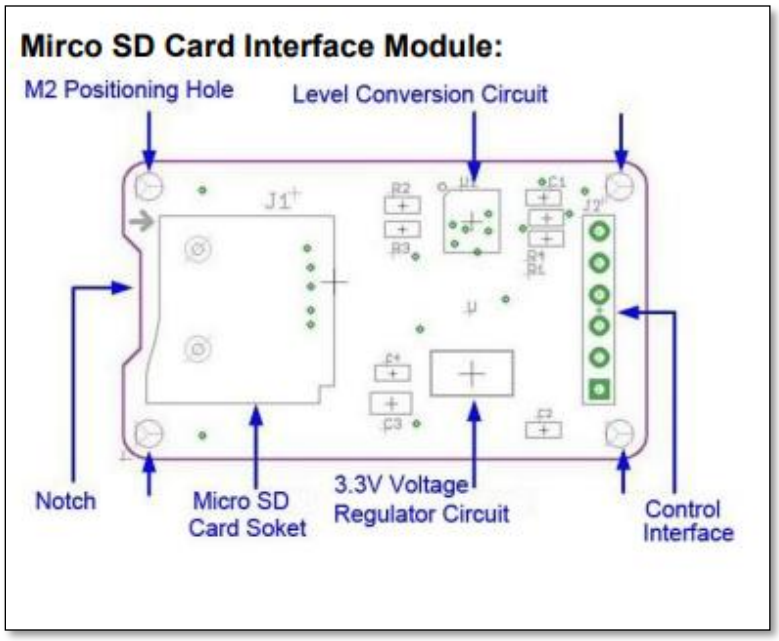
Requirements

It can be interface with any microcontroller such as PIC, SK40C, SK28A, SKds40A, Arduino series.

Necessary hardware to follow this guide:

- Arduino Uno
- Micro SD Card Adapter Reader module
- Male-Female jumper wire

Pin Assignment

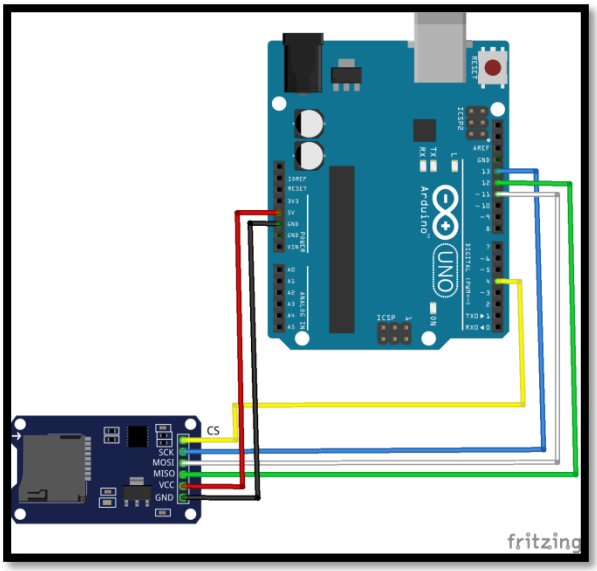


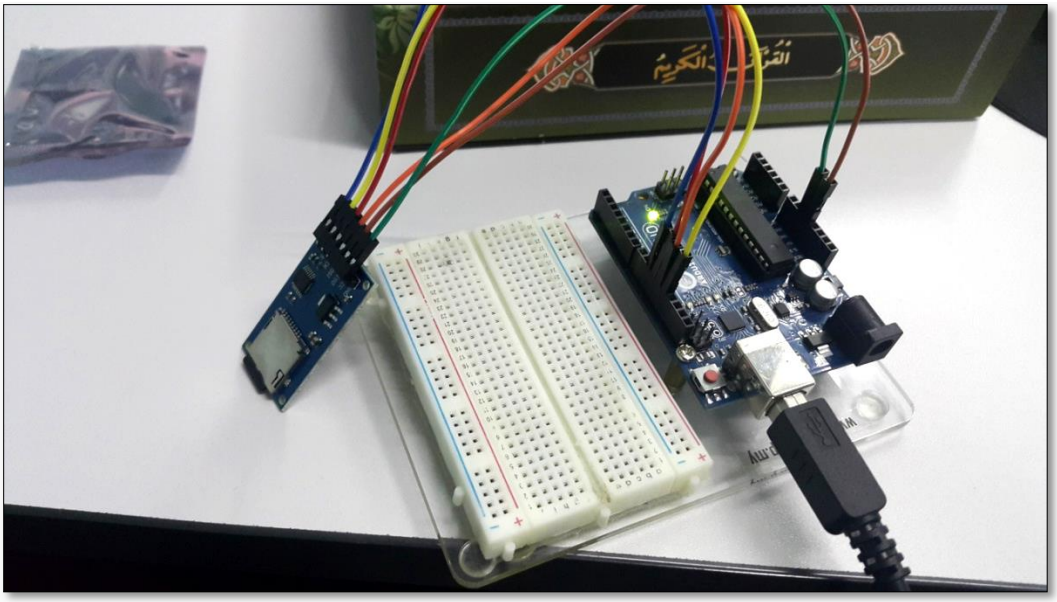
MICROSD CARD ADAPTER PIN	ARDUINO UNO PIN
CS	4
SCK	13
MOSI	11
MISO	12
VCC	5V
GND	GND

Hardware Interface/Setup

The reading and writing an SD card is very simple when the standard SD library of the Arduino IDE is used. Make sure to use the latest version of the SD library (Sketch -> Include Library -> Manage Libraries -> Search for “SD”). For example, in my case, version 1.1.0 did not work with the module. Fortunately, version 1.1.1 did work without any problems. Moreover, the SD card must be formatted as FAT16 or FAT32. If something does not work as expected, a good start for debugging is always to upload CardInfo example of the library (File -> Examples -> SD -> CardInfo) to the Arduino and read the messages of the serial monitor.

In this tutorial's code, a random number between 0 and 9 is written to an SD card. In particular, the number is written to a file named "file.txt". Next, the content of "file.txt" is read. At the end of the loop function, a delay of 5 seconds is added. Please note that when the Arduino is started, it is checked whether a file named "file.txt" exists (see setup function). If so, the file is deleted.





Example Code

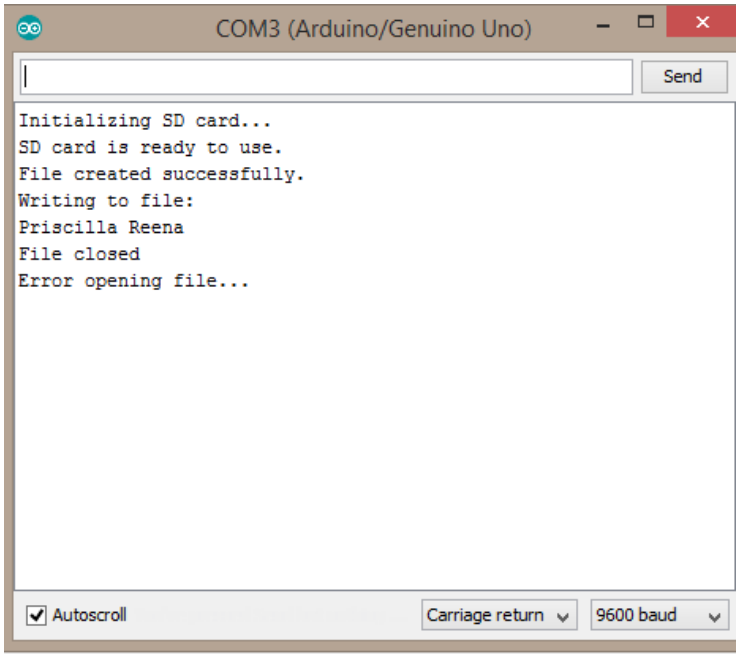
This is example code for moisture sensor module. The full code can be download at http://educ8s.com/Arduino/SDCard/SDCard_demo.ino

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

int CS_PIN = 10;

File file;

void setup()
{

  Serial.begin(9600);

  initializeSD();
  createFile("test.txt");
  writeToFile("This is sample text!");
  closeFile();

  openFile("prefs.txt");
  Serial.println(readLine());
  Serial.println(readLine());
  closeFile();
}
void loop()
{
}

void initializeSD()
{
  Serial.println("Initializing SD card...");
  pinMode(CS_PIN, OUTPUT);

  if (SD.begin())
  {
    Serial.println("SD card is ready to use.");
  } else
  {
    Serial.println("SD card initialization failed");
    return;
  }
}

int createFile(char filename[])
{
```

```
  file = SD.open(filename, FILE_WRITE);

  if (file)
  {
    Serial.println("File created successfully.");
    return 1;
  } else
  {
    Serial.println("Error while creating file.");
    return 0;
  }
}

int writeToFile(char text[])
{
  if (file)
  {
    file.println(text);
    Serial.println("Writing to file: ");
    Serial.println(text);
    return 1;
  } else
  {
    Serial.println("Couldn't write to file");
    return 0;
  }
}

void closeFile()
{
  if (file)
  {
    file.close();
    Serial.println("File closed");
  }
}

int openFile(char filename[])
{
  file = SD.open(filename);
  if (file)
  {
    Serial.println("File opened with success!");
    return 1;
  } else
  {
    Serial.println("Error opening file...");
    return 0;
  }
}
```

```
String readLine()
{
  String received = "";
  char ch;
  while (file.available())
  {
    ch = file.read();
    if (ch == '\n')
    {
      return String(received);
    }
    else
    {
      received += ch;
    }
  }
  return "";
}
```

RESULTS



Applications

1. Water Pump System
2. Automatic Plant-Watering System
3. Plantation Industries
4. DIY Project
5. Microcontroller