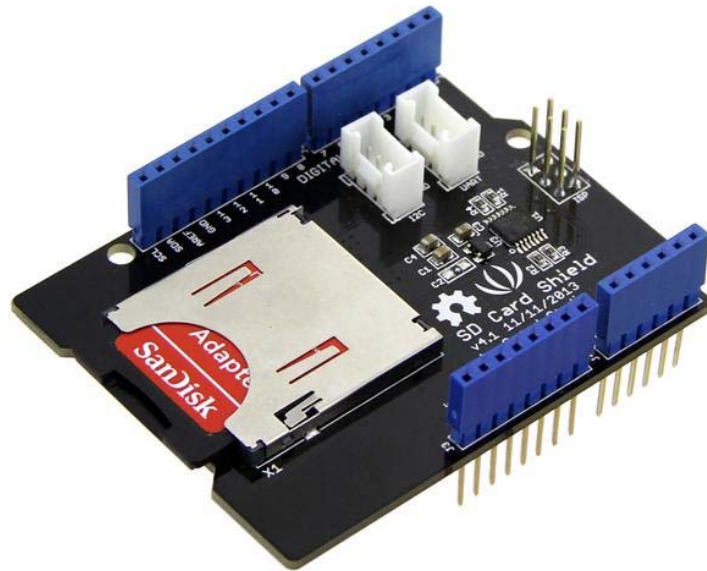# SD Card shield V4.0

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

	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
2.8'' TFT Touch Shield V2.0	bmp nonsupport	bmp nonsupport	Not recommended	bmp nonsupport	Not recommended	Not recommended	Not recommended	Not recommended
Base Shield V2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Camera Shield	Only Pin234567	Hardware Serial OK	Not recommended	Not recommended	Yes	Hardware Serial OK	No	No
EL Shield	Yes	Yes	No	Yes	No	No	No	No
Energy Shield	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
GPRS Shield	Not recommended	Not recommended	Yes	Yes	Yes	Not recommended	Yes	No need
Motor Shield V2.0	Yes	Stepper motor only	No	Yes	Stepper motor only	Stepper motor only	No	No
Music Shield V2.0	Yes	Yes	Not recommended	Yes	Yes	Yes	Yes	Yes
NFC Shield V2.0	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Protoshield Kit for Arduino	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RS232 Shield	Yes	Yes	No	Yes	No	No	No	No
Relay Shield V3.0	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SD Card Shield V4.0	Yes	Yes	Not recommended	Yes	Yes	Yes	No	No
Seeed BLE Shield V1	Yes	Not recommended	Not recommended	Yes	No need	Not recommended	Not recommended	No need
W5500 Ethernet Shield	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Wifi Shield(Fi250) V1.1	Not recommended	Not recommended	Not recommended	Yes	Yes	Not recommended	No need	No need
Wifi Shield V2	Yes	Not recommended	Not recommended	Yes	Yes	Not recommended	No need	No need
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
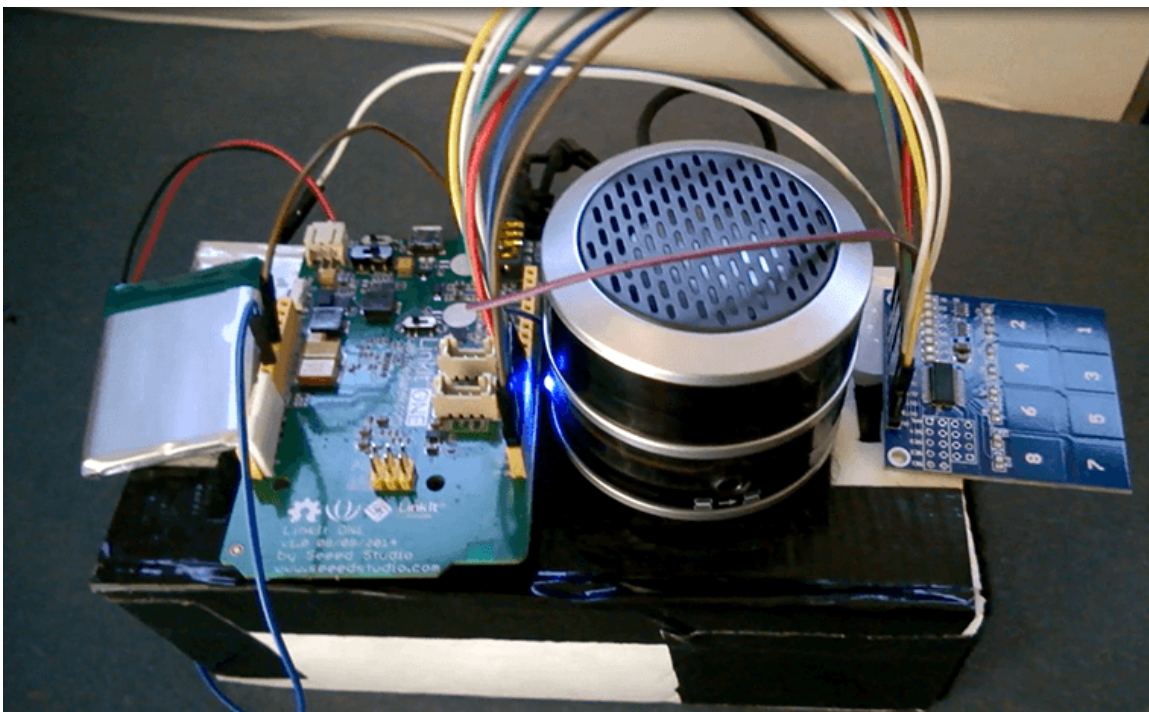
This project uses Grove - Water Sensor to create a simple but effective solution to watering plants.

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

Item	Min	Typical	Max	Unit
Voltage	3.5	5.0	5.5	V
Current	0.159	100	200	mA
Supported Card Type	SD card(<=32GB); Micro SD card(<=32GB); SDHC card(<=32GB)			/
Dimension	68.7x53.5x19.00			mm
Net Weight	14.8			g

Hardware Overview
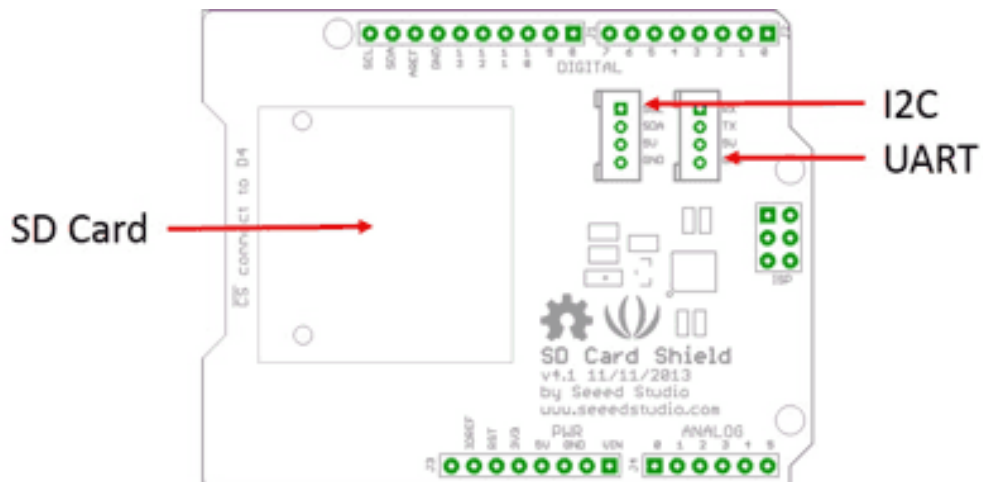
Pins usage on Arduino （with SD card）

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.
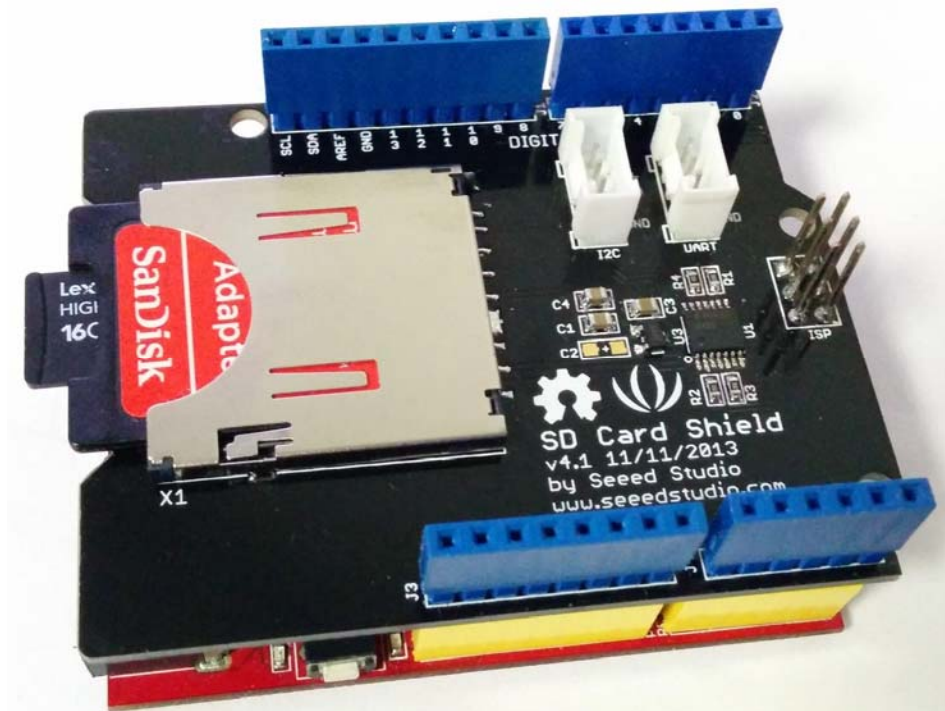
Usage

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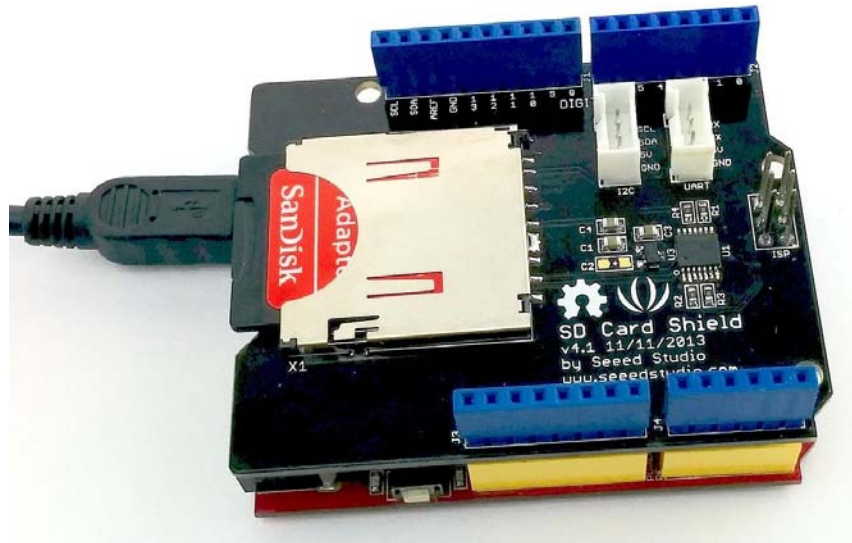
Hardware Installation

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 adapter, 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.
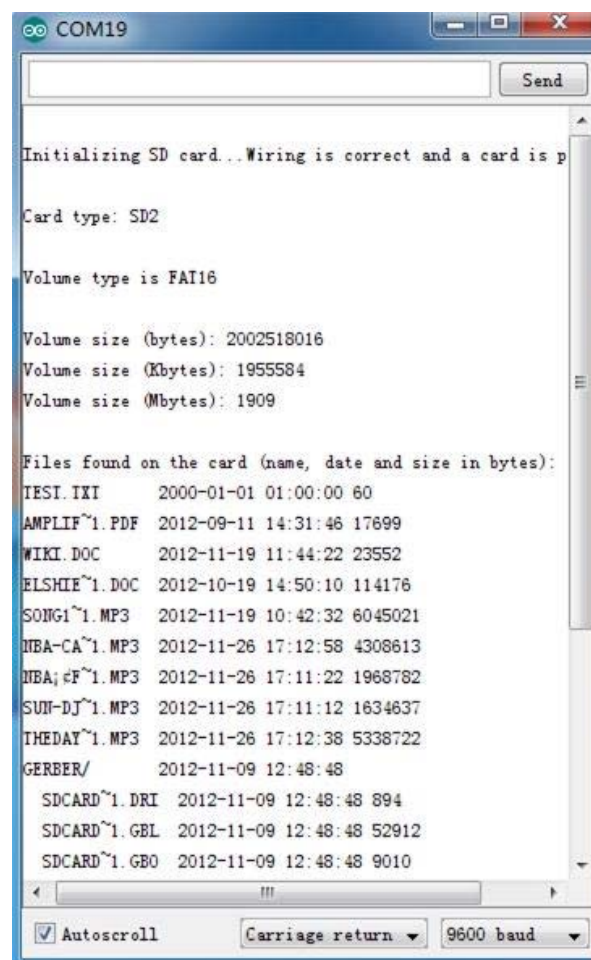
There is a brief description to above code:

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.

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.

```
1    /*
2      SD card test
3
4      This example shows how use the utility libraries on which the'
5      SD library is based in order to get info about your SD card.
6      Very useful for testing a card when you're not sure whether its working or not.
7
8      The circuit:
9       * SD card attached to SPI bus as follows:
10     ** MOSI - pin 11 on Arduino Uno/Duemilanove/Diecimila
11     ** MISO - pin 12 on Arduino Uno/Duemilanove/Diecimila
12     ** CLK - pin 13 on Arduino Uno/Duemilanove/Diecimila
13     ** CS - depends on your SD card shield or module.
14     ** Pin 4 used here for consistency with other Arduino examples
15
16
17     created  28 Mar 2011
18     by Limor Fried
19     modified 9 Apr 2012
20     by Tom Igoe
21     */
22     // include the SD library:
23     #include <SPI.h>
24     #include <SD.h>
25
26     // set up variables using the SD utility library functions:
27     Sd2Card card;
28     SdVolume volume;
29     SdFile root;
30
31     // change this to match your SD shield or module;
32     // Arduino Ethernet shield: pin 4
33     // Adafruit SD shields and modules: pin 10
34     // Sparkfun SD shield: pin 8
35     const int chipSelect = 4;
36
37     void setup()
38     {
39       // Open serial communications and wait for port to open:
40       Serial.begin(9600);
41       while (!Serial) {
42         ; // wait for serial port to connect. Needed for Leonardo only
43       }
44
45
46       Serial.print("\nInitializing SD card...");
47       // On the Ethernet Shield, CS is pin 4. It's set as an output by default.
48       // Note that even if it's not used as the CS pin, the hardware SS pin
49       // (10 on most Arduino boards, 53 on the Mega) must be left as an output
50       // or the SD library functions will not work.
51       pinMode(10, OUTPUT);    // change this to 53 on a mega
52
53
54       // we'll use the initialization code from the utility libraries
55       // since we're just testing if the card is working!
56       if (!card.init(SPI_HALF_SPEED, chipSelect)) {
```

```
57      Serial.println("initialization failed. Things to check:");
58      Serial.println("* is a card is inserted?");
59      Serial.println("* Is your wiring correct?");
60      Serial.println("* did you change the chipSelect pin to match your shield or module?");
61      return;
62    } else {
63      Serial.println("Wiring is correct and a card is present.");
64    }
65
66    // print the type of card
67    Serial.print("\nCard type: ");
68    switch (card.type()) {
69      case SD_CARD_TYPE_SD1:
70        Serial.println("SD1");
71        break;
72      case SD_CARD_TYPE_SD2:
73        Serial.println("SD2");
74        break;
75      case SD_CARD_TYPE_SDHC:
76        Serial.println("SDHC");
77        break;
78      default:
79        Serial.println("Unknown");
80    }
81
82    // Now we will try to open the 'volume'/'partition' - it should be FAT16 or FAT32
83    if (!volume.init(card)) {
84      Serial.println("Could not find FAT16/FAT32 partition.\nMake sure you've formatted the card");
85      return;
86    }
87
88
89    // print the type and size of the first FAT-type volume
90
91    uint64_t volumesize64;
92    uint32_t volumesize32;
93    Serial.print("\nVolume type is FAT");
94    Serial.println(volume.fatType(), DEC);
95    Serial.println();
96
97    volumesize64 = volume.blocksPerCluster();    // clusters are collections of blocks
98    volumesize64 *= volume.clusterCount();        // we'll have a lot of clusters
99    volumesize64 *= 512;                          // SD card blocks are always 512 bytes
100
101    Serial.print("Volume size (bytes): ");
102    printLLNumber(volumesize64, DEC);
103    Serial.println();
104
105    Serial.print("Volume size (Kbytes): ");
106    volumesize32 = volumesize64/1024;
107    Serial.println(volumesize32);
108
109    Serial.print("Volume size (Mbytes): ");
110    volumesize32 /= 1024;
111    Serial.println(volumesize32);
112    /*uint64_t volumesize;
```

```
113    Serial.print("\nVolume type is FAT");
114    Serial.println(volume.fatType(), DEC);
115    Serial.println();
116
117    volumesize = volume.blocksPerCluster();    // clusters are collections of blocks
118    volumesize *= volume.clusterCount();       // we'll have a lot of clusters
119    volumesize *= 512;                          // SD card blocks are always 512 bytes
120    Serial.print("Volume size (bytes): ");
121    Serial.println(volumesize,DEC);
122    Serial.print("Volume size (Kbytes): ");
123    volumesize /= 1024;
124    Serial.println(volumesize,DEC);
125    Serial.print("Volume size (Mbytes): ");
126    volumesize /= 1024;
127    Serial.println(volumesize,DEC);
128  */
129
130    Serial.println("\nFiles found on the card (name, date and size in bytes): ");
131    root.openRoot(volume);
132
133    // list all files in the card with date and size
134    root.ls(LS_R | LS_DATE | LS_SIZE);
135  }
136
137
138  void loop(void) {
139
140  }
141  void printLLNumber(uint64_t n, uint8_t base)
142  {
143    unsigned char buf[16 * sizeof(long)];
144    unsigned int i = 0;
145
146    if (n == 0)
147    {
148      Serial.print((char)'0');
149      return;
150    }
151
152    while (n > 0)
153    {
154      buf[i++] = n % base;
155      n /= base;
156    }
157
158    for (; i > 0; i--)
159      Serial.print((char) (buf[i - 1] < 10 ?
160        '0' + buf[i - 1] :
161        'A' + buf[i - 1] - 10));
162  }
```

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.