

## Code

This example seems to get installed when you add support for ESP32 boards to the arduino IDE – the standard Arduino sd card example does not work

```
1  /*
2  * Connect the SD card to the following pins:
3  *
4  * SD Card | ESP32
5  * D2 -
6  * D3 SS
7  * CMD MOSI
8  * VSS GND
9  * VDD 3.3V
10 * CLK SCK
11 * VSS GND
12 * D0 MISO
13 * D1 -
14 */
15 #include "FS.h"
16 #include "SD.h"
17 #include "SPI.h"
18
19 void listDir(fs::FS &fs, const char * dirname, uint8_t levels){
20   Serial.printf("Listing directory: %s\n", dirname);
21
22   File root = fs.open(dirname);
23   if(!root){
24     Serial.println("Failed to open directory");
25     return;
26   }
27   if(!root.isDirectory()){
28     Serial.println("Not a directory");
29     return;
30   }
31 }
```

```
32 File file = root.openNextFile();
33 while(file){
34     if(file.isDirectory()){
35         Serial.print(" DIR : ");
36         Serial.println(file.name());
37         if(levels){
38             listDir(fs, file.name(), levels -1);
39         }
40     } else {
41         Serial.print(" FILE: ");
42         Serial.print(file.name());
43         Serial.print(" SIZE: ");
44         Serial.println(file.size());
45     }
46     file = root.openNextFile();
47 }
48 }
49
50 void createDir(fs::FS &fs, const char * path){
51     Serial.printf("Creating Dir: %s\n", path);
52     if(fs.mkdir(path)){
53         Serial.println("Dir created");
54     } else {
55         Serial.println("mkdir failed");
56     }
57 }
58
59 void removeDir(fs::FS &fs, const char * path){
60     Serial.printf("Removing Dir: %s\n", path);
61     if(fs.rmdir(path)){
62         Serial.println("Dir removed");
63     } else {
64         Serial.println("rmdir failed");
65     }
66 }
67
68 void readFile(fs::FS &fs, const char * path){
69     Serial.printf("Reading file: %s\n", path);
70
71     File file = fs.open(path);
72     if(!file){
73         Serial.println("Failed to open file for reading");
74         return;
75     }
76
77     Serial.print("Read from file: ");
```

```
78 while(file.available()){
79     Serial.write(file.read());
80 }
81 }
82
83 void writeFile(fs::FS &fs, const char * path, const char * message){
84     Serial.printf("Writing file: %s\n", path);
85
86     File file = fs.open(path, FILE_WRITE);
87     if(!file){
88         Serial.println("Failed to open file for writing");
89         return;
90     }
91     if(file.print(message)){
92         Serial.println("File written");
93     } else {
94         Serial.println("Write failed");
95     }
96 }
97
98 void appendFile(fs::FS &fs, const char * path, const char * message){
99     Serial.printf("Appending to file: %s\n", path);
100
101     File file = fs.open(path, FILE_APPEND);
102     if(!file){
103         Serial.println("Failed to open file for appending");
104         return;
105     }
106     if(file.print(message)){
107         Serial.println("Message appended");
108     } else {
109         Serial.println("Append failed");
110     }
111 }
112
113 void renameFile(fs::FS &fs, const char * path1, const char * path2){
114     Serial.printf("Renaming file %s to %s\n", path1, path2);
115     if (fs.rename(path1, path2)) {
116         Serial.println("File renamed");
117     } else {
118         Serial.println("Rename failed");
119     }
120 }
121
122 void deleteFile(fs::FS &fs, const char * path){
123     Serial.printf("Deleting file: %s\n", path);
```

```
124     if(fs.remove(path)){
125         Serial.println("File deleted");
126     } else {
127         Serial.println("Delete failed");
128     }
129 }
130
131 void testFileIO(fs::FS &fs, const char * path){
132     File file = fs.open(path);
133     static uint8_t buf[512];
134     size_t len = 0;
135     uint32_t start = millis();
136     uint32_t end = start;
137     if(file){
138         len = file.size();
139         size_t flen = len;
140         start = millis();
141         while(len){
142             size_t toRead = len;
143             if(toRead > 512){
144                 toRead = 512;
145             }
146             file.read(buf, toRead);
147             len -= toRead;
148         }
149         end = millis() - start;
150         Serial.printf("%u bytes read for %u ms\n", flen, end);
151         file.close();
152     } else {
153         Serial.println("Failed to open file for reading");
154     }
155
156     file = fs.open(path, FILE_WRITE);
157     if(!file){
158         Serial.println("Failed to open file for writing");
159         return;
160     }
161
162     size_t i;
163     start = millis();
164     for(i=0; i<2048; i++){
165         file.write(buf, 512);
166     }
167     end = millis() - start;
168     Serial.printf("%u bytes written for %u ms\n", 2048 * 512, end);
169     file.close();
```

```
170 }
171
172 void setup(){
173   Serial.begin(115200);
174   if(!SD.begin()){
175     Serial.println("Card Mount Failed");
176     return;
177   }
178   uint8_t cardType = SD.cardType();
179
180   if(cardType == CARD_NONE){
181     Serial.println("No SD card attached");
182     return;
183   }
184
185   Serial.print("SD Card Type: ");
186   if(cardType == CARD_MMC){
187     Serial.println("MMC");
188   } else if(cardType == CARD_SD){
189     Serial.println("SDSC");
190   } else if(cardType == CARD_SDHC){
191     Serial.println("SDHC");
192   } else {
193     Serial.println("UNKNOWN");
194   }
195
196   uint64_t cardSize = SD.cardSize() / (1024 * 1024);
197   Serial.printf("SD Card Size: %lluMB\n", cardSize);
198
199   listDir(SD, "/", 0);
200   createDir(SD, "/mydir");
201   listDir(SD, "/", 0);
202   removeDir(SD, "/mydir");
203   listDir(SD, "/", 2);
204   writeFile(SD, "/hello.txt", "Hello ");
205   appendFile(SD, "/hello.txt", "World!\n");
206   readFile(SD, "/hello.txt");
207   deleteFile(SD, "/foo.txt");
208   renameFile(SD, "/hello.txt", "/foo.txt");
209   readFile(SD, "/foo.txt");
210   testFileIO(SD, "/test.txt");
211 }
212
213 void loop(){
214
215 }
```

## Output

Open the serial monitor – this was my sample micro sd card

```
SD Card Type: SDSC
SD Card Size: 241MB
Listing directory: /
FILE: /MCP9808.TXT SIZE: 1496
FILE: /HDC1000.CSV SIZE: 836
FILE: /test.txt SIZE: 0
FILE: /foo.txt SIZE: 13
DIR : /System Volume Information
FILE: /miniwoof.bmp SIZE: 57654
FILE: /test.bmp SIZE: 230456
FILE: /woof.bmp SIZE: 230456
Creating Dir: /mydir
Dir created
Listing directory: /
FILE: /MCP9808.TXT SIZE: 1496
FILE: /HDC1000.CSV SIZE: 836
FILE: /test.txt SIZE: 0
FILE: /foo.txt SIZE: 13
DIR : /mydir
DIR : /System Volume Information
FILE: /miniwoof.bmp SIZE: 57654
FILE: /test.bmp SIZE: 230456
FILE: /woof.bmp SIZE: 230456
Removing Dir: /mydir
```

Dir removed

Listing directory: /

FILE: /MCP9808.TXT SIZE: 1496

FILE: /HDC1000.CSV SIZE: 836

FILE: /test.txt SIZE: 0

FILE: /foo.txt SIZE: 13

DIR : /System Volume Information

Listing directory: /System Volume Information

FILE: /System Volume Information/IndexerVolumeGuid SIZE: 76

FILE: /miniwoof.bmp SIZE: 57654

FILE: /test.bmp SIZE: 230456

FILE: /woof.bmp SIZE: 230456

Writing file: /hello.txt

File written

Appending to file: /hello.txt

Message appended

Reading file: /hello.txt

Read from file: Hello World!

Deleting file: /foo.txt

File deleted

Renaming file /hello.txt to /foo.txt

File renamed

Reading file: /foo.txt

Read from file: Hello World!

0 bytes read for 0 ms

1048576 bytes written for 18725 ms

## Links