SDCARDFILES:

This class was written for the Arduino library.

It creates text files and appends data to them.
It's use is for mass storage on the arduino platform.
This data is then loaded onto and used on a PC.

Import the library:

```
#include <SDCARDFILES.h>
```

It has only two functions:

```
SDCARDFILES.createfile();  //create a text file with 0 length
SDCARDFILES.fileappend();  //append 512 bytes to the open file
```

The 512 bytes are written from a buffer which must be declared as global:

```
unsigned char buffer[512] ; //contains the 512 bytes to write to file
```

Operational notes:

This library is based on the SRAM library :

http://arduino.cc/forum/index.php/topic,52871.0.html

It works on standard SD cards from 256MB to 2GB made between 2007-2011.

Start with a newly formatted(FAT16) card.

The first call of createfile will create a text file in the first root directory location:

DATA0000.TXT   it will have length = 0

Further calls will create sequential number files up to the maximum allowed in the root directory(usually 512):

DATA00511.TXT   it will have length = 0

Calls to fileappend will write the 512 byte buffer to the end of the currently opened file.
Further calls will append until the data sectors are used up.

Both createfile and fileappend use the 512 byte buffer.
It will be changed after the call.

You should use multiples of the cluster size to use the card efficiently.

To help with this and other decisions about your program use the example SDCARDbootrecord to get this information(2Gb card):

```
bootsector =  251
partitionsectors =  3962629
bytespersector =  512
sectorspercluster =  64
reservedsectors =  1
fatcopies =  2
rootdirnumber =  512
sectorsperFAT =  242
totalfilesectors =  3962629

FAT1start =  252
FAT2start =  494
rootdirectorystart =  736
datastartsector =  768
clusterbytes =  32768
maximumdataclusters =  61904
maximum clusters per file spread over all files =  120
```

The class uses 70 bytes of RAM, 2960 bytes of program FLASH.
Of course you must use 512 bytes of RAM for the buffer.

On average a file append takes 60 ms.

Master Boot Record:

For the following sector dumps use the example program SDCARDdump.
It dumps a sector 32 bytes per line.

The master boot record at sector 0 is sometimes called the partition sector.
This is because it contains the partition table for the card.

The sector in raw numbers:

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3
63  0  6  54  246  214  251  0  0  0  5  119  60  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 85  170
```

The last 2 bytes 170 85 is MBR signature 0x55AA.

The 4 bytes at offset 0x1C6(454)   251 0 0 0 is the boot record address = 251.
The 4 bytes at offset 0x1CA(458) 5 119 60 0 is the number of sectors
in the partition = 3962629.

SD cards are sold formatted(FAT16) with a master boot record. If we format again using windows the MBR will be retained.
If the MBR(sector 0) has been written over a windows format will not replace it.
The boot record will be placed at sector 0.

Boot Record:

This is the boot record at sector at sector 251:

```
The sector in raw numbers:
235  0  144  32  32  32  32  32  32  32  32  0  2  64  1  0  2  0  2  0  0  248  242  0  63  0  64  0  251  0  0  0
5  119  60  0  128  0  41  169  61  48  252  78  79  32  78  65  77  69  32  32  32  32  70  65  84  49  54  32  32  32  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  85  170
```

The last 2 bytes 170 85 is the boot record signature 0x55AA.

The first byte is always 235 (was a jump instruction).

The entries we are interested in are:

```
int bytespersector; //is at offset 11
char sectorspercluster;  //is at offset 13
int reservedsectors;  //is at offset 14
char fatcopies;  //is at offset 16
int rootdirnumber;  //is at offset 17

int sectorsperFAT;  //is at offset 22

long totalfilesectors;  //is at offset 32
```

Root file directory:

Createfile writes text file entries into the root file directory(starts sector 736).

This is after creating 16 files of length 512 bytes:

The sector in raw numbers:

```
68 65 84 65 48 48 48 48 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 0 0
68 65 84 65 48 48 48 49 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 2 0 0
68 65 84 65 48 48 48 50 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 2 0 0
68 65 84 65 48 48 48 51 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 2 0 0
68 65 84 65 48 48 48 52 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 2 0 0
68 65 84 65 48 48 48 53 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 2 0 0
68 65 84 65 48 48 48 54 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 2 0 0
68 65 84 65 48 48 48 55 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 2 0 0
68 65 84 65 48 48 48 56 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 2 0 0
68 65 84 65 48 48 48 57 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 11 0 0 2 0 0
68 65 84 65 48 48 49 48 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 2 0 0
68 65 84 65 48 48 49 49 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 13 0 0 2 0 0
68 65 84 65 48 48 49 50 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 14 0 0 2 0 0
68 65 84 65 48 48 49 51 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 15 0 0 2 0 0
68 65 84 65 48 48 49 52 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 16 0 0 2 0 0
68 65 84 65 48 48 49 53 84 88 84 32 16 0 0 0 0 0 0 0 0 0 0 0 0 0 17 0 0 2 0 0
```

Each file has 32 bytes(one line in dump example).

The first 11 bytes are the file name.
The last 4 bytes are a long number with the length of the file(512).
The preceeding 2 bytes are an integer number giving the start cluster
for the file(the first cluster to be used = 2).

File allocation table:

The FAT is an integer(2bytes) listing of the clusters used by each file.

0x0000 Free cluster
0x0001 Reserved cluster
0x0002 - 0xFFEF Used cluster; value points to next cluster
0xFFF0 - 0xFFF6 Reserved values
0xFFF7 Bad cluster
0xFFF8 - 0xFFFF Last cluster in file

This is the FAT after creating DATA0000.txt with 0 length.

The sector in raw numbers:

```
248  255  255  255  255  255  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

The first cluster to be used is 2.
The first 2(0 and 1) are always shown to be last in the file.
248 255_255 255_255 255 ....here cluster 2 is the last cluster for DATA0000.

This is the FAT after creating DATA0000.txt with 128 sectors(2clusters) length.

The sector in raw numbers:

```
248  255  255  255  3  0  255  255  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

248 255_255 255_3 0_255 255 ....here cluster 2 shows cluster 3 is the next cluster used by the file and cluster 3 is the last cluster for DATA0000.

Example programs:

These programs write to the SD card.

SDCARDwritesector: writes one sector on the SD card

Create512files: create 512 text files   DATA0000.TXT - DATA0511.TXT
each file is one sector(512 bytes) long.