

Verwendung von SD-Speicherkarten über den SPI-Bus

- Eigenschaften von SD-Speicherkarten
- File Allocation Table (FAT)
- Ansteuerung von SD-Karten
- Karten-Interface-Board
- Schaltung des SD-Experiments
- Software
- Anhang: Formatieren einer SD-Karte unter Windows XP

Eigenschaften von SD-Speicherkarten

SD-Speicherkarten sind gängige Speichermedien mit erheblichen Kapazitäten. Derzeit liegt die Kapazität käuflicher Karten zwischen 2 und 32 GB.

Diese Karten (für PC-Nutzung) sind intern wie Festplatten organisiert und können deswegen auch leicht auf dem PC zu lesen. Man benötigt dazu einen Kartenadapter, der bei manchen PCs bereits eingebaut ist. Es gibt aber auch USB-Adapter, die sich nach Einstecken als Laufwerk im PC melden.

Vor Gebrauch muss man die Karten wie Festplatten formatieren. SD-Karten aus Fotoaparaten und anderen Geräten müssen nicht zwingend genauso formatiert sein.



Beispiel eines USB-Kartenadapters



MicroSD-Karte
+ SD Adapter

Quelle: <http://www.ladyada.net/products/microsd/>

Die MicroSD-Karten können über einen Adapter auch in Sockel für normale SD-Karten gesteckt werden

File Allocation Table (FAT)

Die Karten werden beim Formatieren mit einer **FAT (File Allocation Table)** versehen, die die Zuordnung der physikalischen Speicherstellen zu Datenfiles festlegt (siehe Anhang).

Man kann beim Formatieren aussuchen welches System man wählt. Die Arduino-Library versteht z.B. FAT16 und FAT32. Bei heutigen Größen der SD-Karten empfiehlt sich die FAT32.

Quelle: http://de.wikipedia.org/wiki/File_Allocation_Table#FAT12

FAT16 hat folgende Merkmale:

- 65.536 Einträge sind möglich, allerdings nur 512 im Rootverzeichnis.
- Es können $2^{16} - 12 = 65.524$ Cluster angesprochen werden (12 Cluster reserviert FAT16, deshalb nicht 65.536).
- 8.3-Dateinamensformat (z.B. ABCDEFGH.TXT)
- Die Partitionsgröße ist unter MS-DOS und Windows 9x (die auf DOS basieren) auf 2 GB begrenzt.
- Die Cluster sind je nach Partitionsgröße zwischen 512 Byte und maximal 32 kB (Windows NT maximal 64 kB) groß.
- Dateien dürfen bis 2 GB groß werden, unter Windows NT.
- Das Haupt-(Root-)Verzeichnis muss sich an einer bestimmten Position auf dem Datenträger befinden. Seine Größe wird bei Erstellung des Dateisystems festgelegt, d. h. es kann nicht wachsen.

FAT32 hat folgende Merkmale:

- Es werden bis zu $2^{28} = 268.435.456$ Cluster verwendet.
- Das Dateisystem ist auf ca. 8,8 Terabyte begrenzt
- Die Cluster sind je nach Partitionsgröße zwischen 512 Byte und maximal 32 kB groß.
- Dateien dürfen max. bis zu 4 GB – 1 Byte (= 4.294.967.295 Byte) groß werden.
- Jede Datei belegt mindestens einen Cluster.
- Das Haupt-(Root-)Verzeichnis muss sich nicht an einer bestimmten Position auf dem Datenträger befinden und hat vor allem keine festgelegte Größe.
- Alte DOS-/Windows-Versionen können nicht auf FAT32 zugreifen.
- Unter Windows 2000 und Nachfolgern darf der Benutzer mit der eingebauten Funktion „Formatieren“ maximal 32 GB große FAT32-Dateisysteme neu erstellen. Auch das Kommandozeilen-Programm format.com hat diese Beschränkung.
- Es gibt mehrere moderne DOS-Systeme, die FAT32 nativ unterstützen.

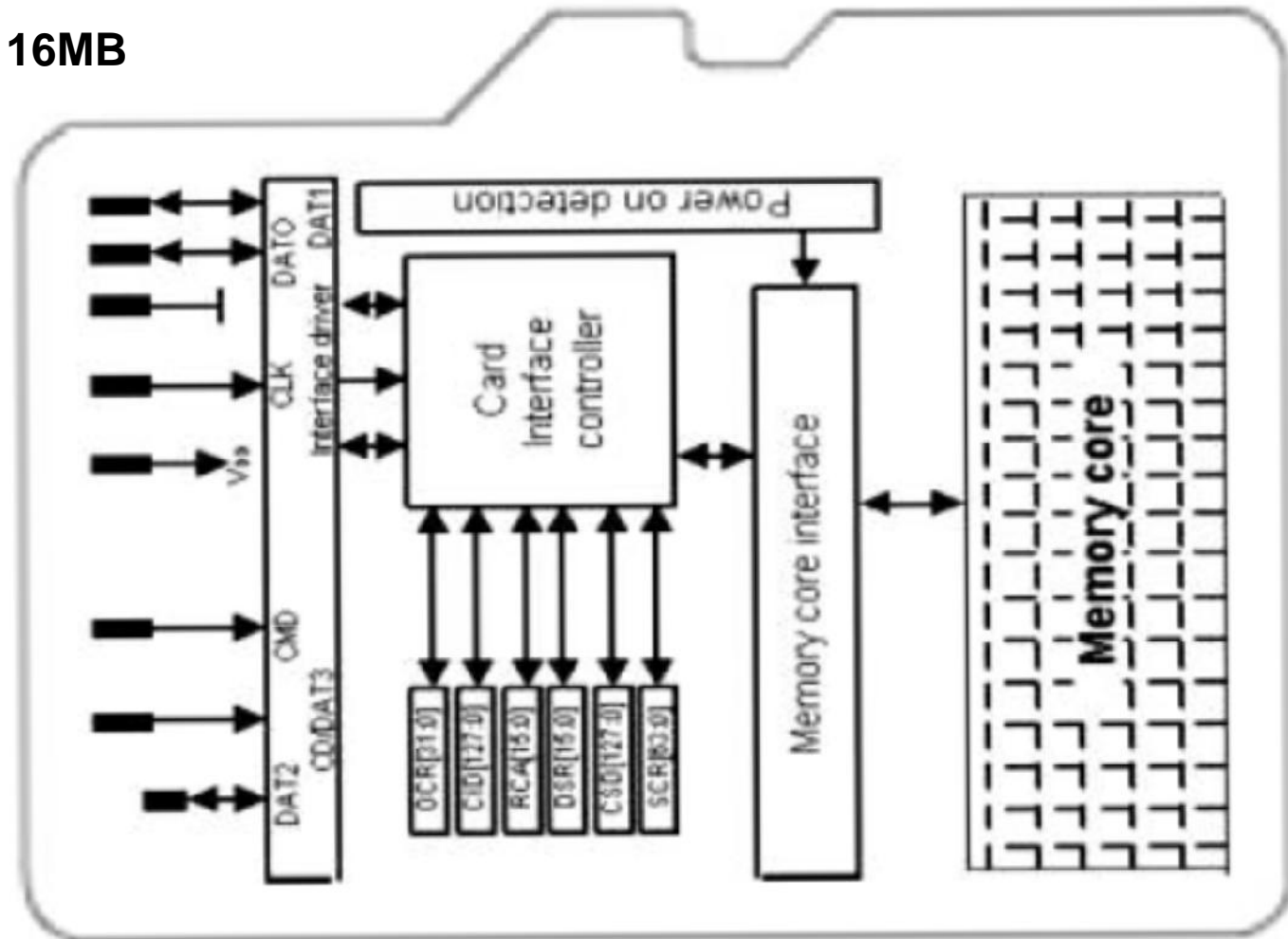
Ansteuerung von SD-Karten



microSDHC Card series

1 2 3 4 5 6 7 8

16MB



Pin Definition

Pin No.	SD Mode			SPI Mode		
	Name	Type	Description	Name	Type	Description
1	DAT2	I/O/PP	Data Line [Bit2]	RSV		Reserved
2	CD/DAT3	I/O/PP	Card Detect / Data Line [Bit3]	CS	I	Chip Select
3	CMD	PP	Command / Response	DI	I	Data In
4	V _{DD}	S	Supply voltage	V _{DD}	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	V _{SS}	S	Supply voltage ground	V _{SS}	S	Supply voltage ground
7	DAT0	I/O/PP	Data Line [Bit0]	DO	O/PP	Data out
8	DAT1	I/O/PP	Data Line [Bit1]	RSV		Reserved

S: Power Supply; I:Input; O:Output; PP:Push-Pull

• Power Supply Voltage

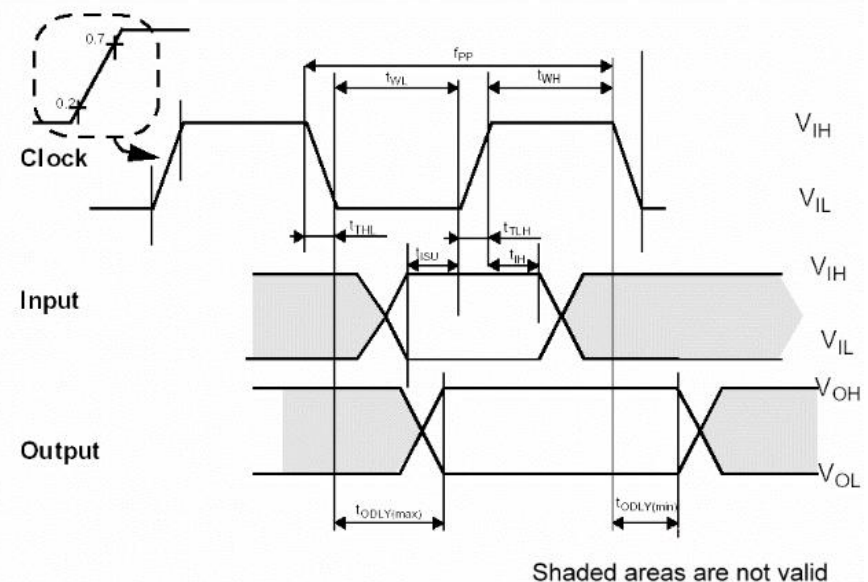
Parameter	Symbol	Min.	Max.	Unit	Remark
Supply voltage	V_{DD}	2.7	3.6	V	
Output High Voltage	V_{OH}	$0.75 \cdot V_{DD}$		V	$I_{OH} = -100 \mu A @ V_{DD} \text{ Min.}$
Output Low Voltage	V_{OL}		$0.125 \cdot V_{DD}$	V	$I_{OL} = 100 \mu A @ V_{DD} \text{ Min.}$
Input High Voltage	V_{IH}	$0.625 \cdot V_{DD}$	$V_{DD} + 0.3$	V	
Input Low Voltage	V_{IL}	$V_{SS} - 0.3$	$0.25 \cdot V_{DD}$	V	
Power up time			250	ms	From 0v to $V_{DD} \text{ Min.}$

• Current Consumption

The current consumption is measured by averaging over 1 s

- Before first command: Maximum 15 mA
- During initialization: Maximum 100 mA
- Operation in Default Mode: Maximum 100 mA
- Operation in High Speed Mode: Maximum 200 mA
- Operation with other functions: Maximum 500 mA.

• Bus Timing

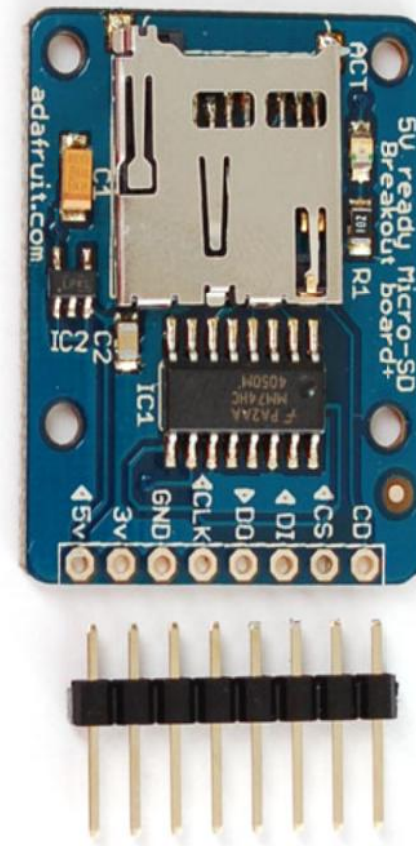
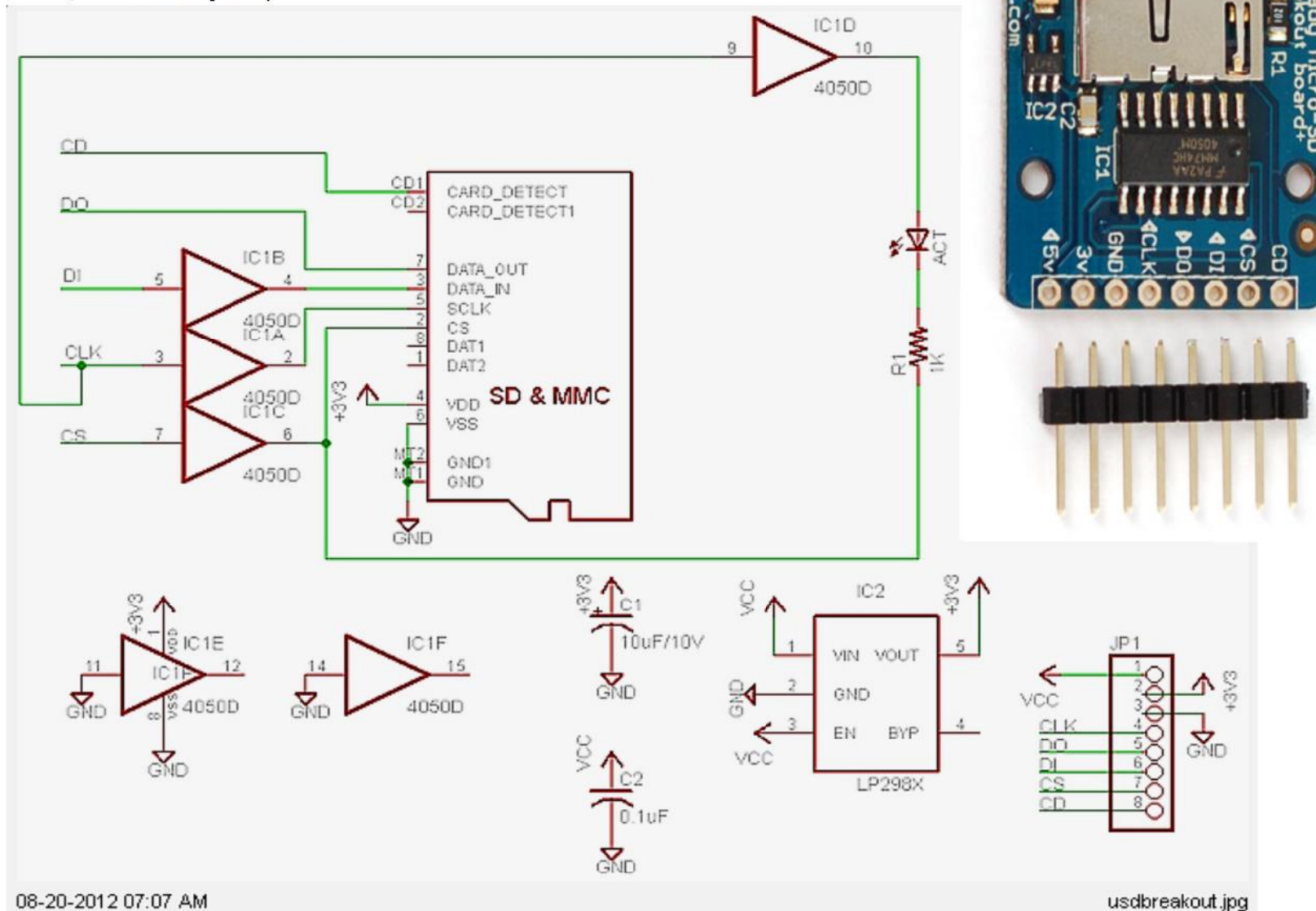


Parameter	Symbol	Min	Max.	Unit	Remark
Clock CLK (All values are referred to min (V_{IH}) and max (V_{IL}))					
Clock frequency Data Transfer Mode	f_{PP}	0	25	MHz	$C_{CARD} \leq 10 \text{ pF, (1 card)}$
Clock frequency Identification Mode	f_{ID}	$0_{min}/100$	400	KHz	$C_{CARD} \leq 10 \text{ pF, (1 card)}$

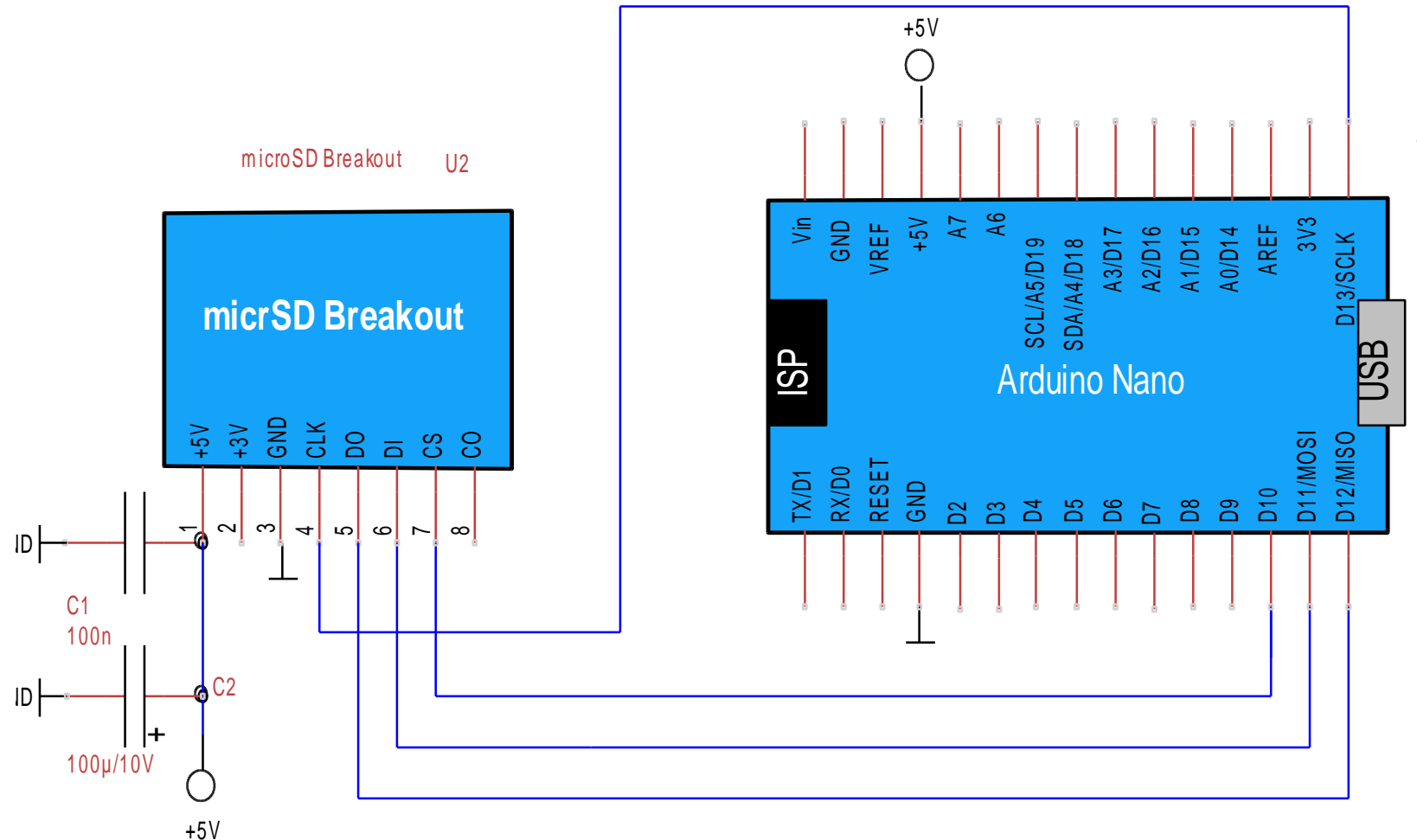
Karten-Interface-Board

SD-Karten Breakout

(Quelle: Ladyada)

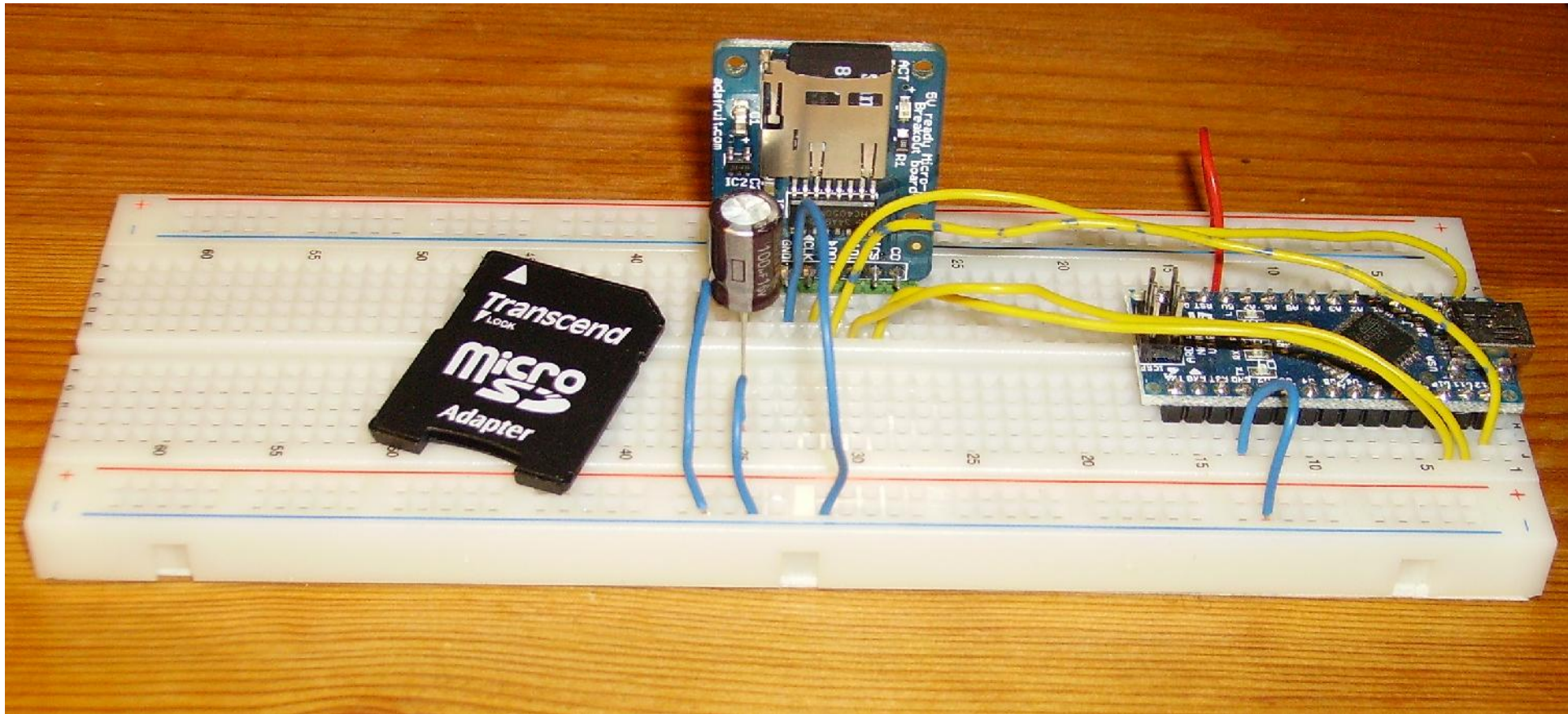


Schaltung des SD-Experiments



Achtung ! Spitzenströme bis 0,2A !
Externes Netzteil verwenden !

Karte während des Schreibens nicht entfernen, Zerstörungsgefahr !



Software

SD_ReadWriteExp10 (1)

```
// SD Karte Schreiben / Lesen
// Minimal-Version zur
//prinzipiellen Darstellung

// MOSI - pin 11
// MISO - pin 12
// CLK - pin 13
// CS - pin 10

#include <SD.h>

#define cs 10
File myFile;

void setup()
{
  // initialisierung des Seriellen Monitors
  Serial.begin(9600);

  pinMode(cs, OUTPUT);
```

SD_ReadWriteExp10 (2)

```
// Ist SD-Karte eingesteckt ?
if (!SD.begin(cs)) {

    return;
}

myFile = SD.open("test.txt", FILE_WRITE);

// Wenn das File geöffnet ist, dann kann geschrieben werden
if (myFile)
{
    myFile.println("the quick brown fox jumps over the lazy dog 1234567890");
    myFile.close();
}
else
{
    // Fehlermeldung
}
```

SD_ReadWriteExp10 (3)

```
myFile = SD.open("test.txt");

if (myFile)
{

    // Lesen vom File und Ausgabe bis nichts mehr drin ist
    while (myFile.available()) {
        Serial.write(myFile.read());
    }
    myFile.close();
}
else
{
    // Fehlermeldung
}
}

void loop ()
{

}
```

Textausgabe des Programms SPI_SD_Exp9

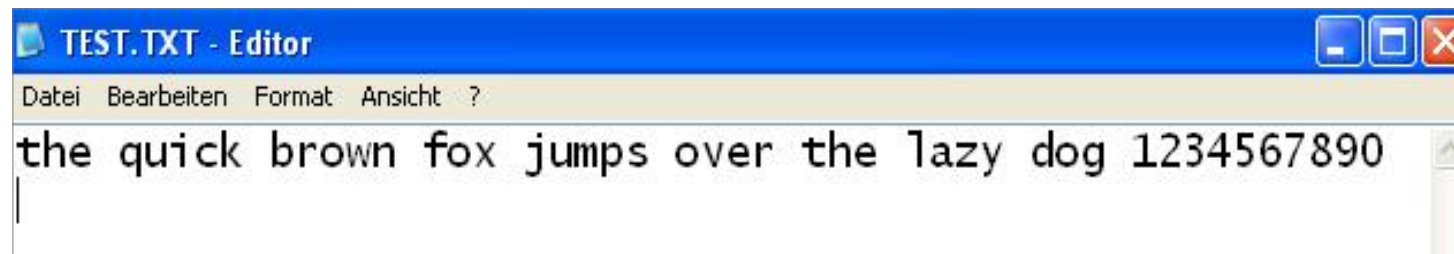
```
Initialisierung SD Karte...Initialisierung gelungen
Altes File loeschen

Oeffnen File test.txt
Schreiben in File test.txt:
the quick brown fox jumps over the lazy dog 1234567890
Schreiben Ende, File schliessen

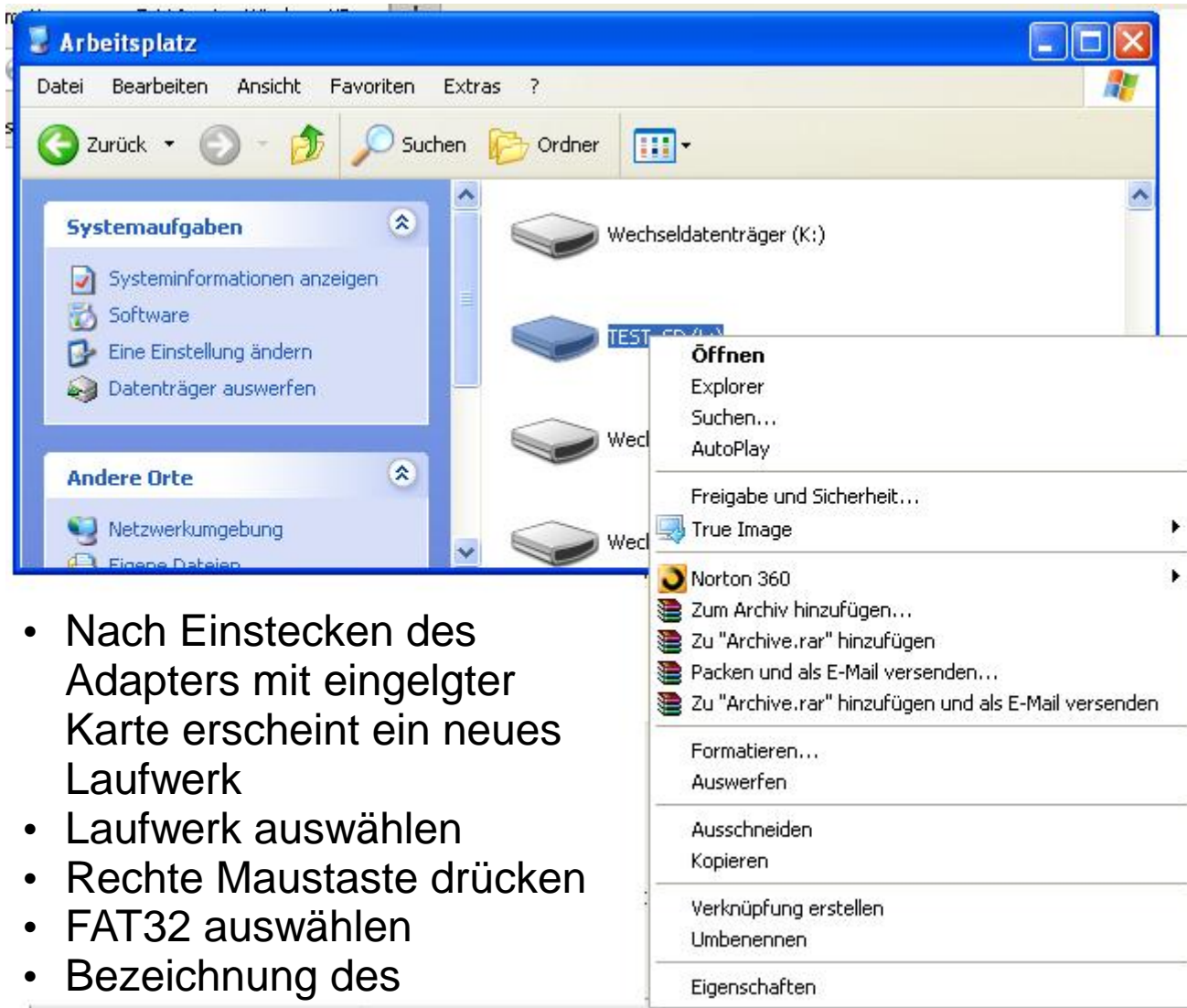
Oeffnen File test.txt
Die Laenge von test.txt ist: 27
test.txt:
the quick brown fox jumps over the lazy dog 1234567890
File test.txt schliessen
```



**Verzeichnis der SD-Karte
und Inhalt über PC
gelesen**



Anhang: Formatieren einer SD-Karte unter Windows XP



- Nach Einstecken des Adapters mit eingelgter Karte erscheint ein neues Laufwerk
- Laufwerk auswählen
- Rechte Maustaste drücken
- FAT32 auswählen
- Bezeichnung des Datenträgers eingeben
- starten

TIP Linux. mkfs.vfat -F 16

- Windows Explorer starten

Das Verfahren sollte auch mit neueren WINDOWS-System ähnlich funktionieren

