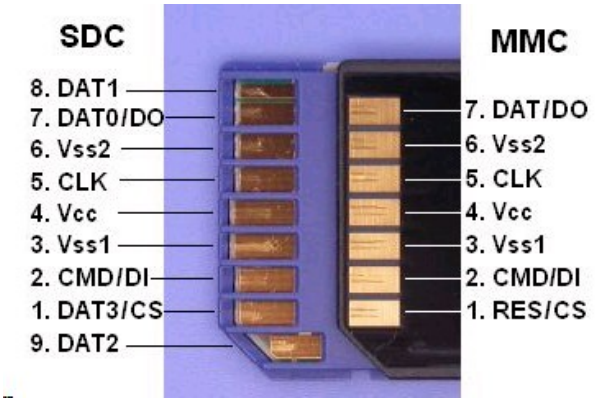
## *Speicherkarten (SD) am AVR-Mikrokontroller*

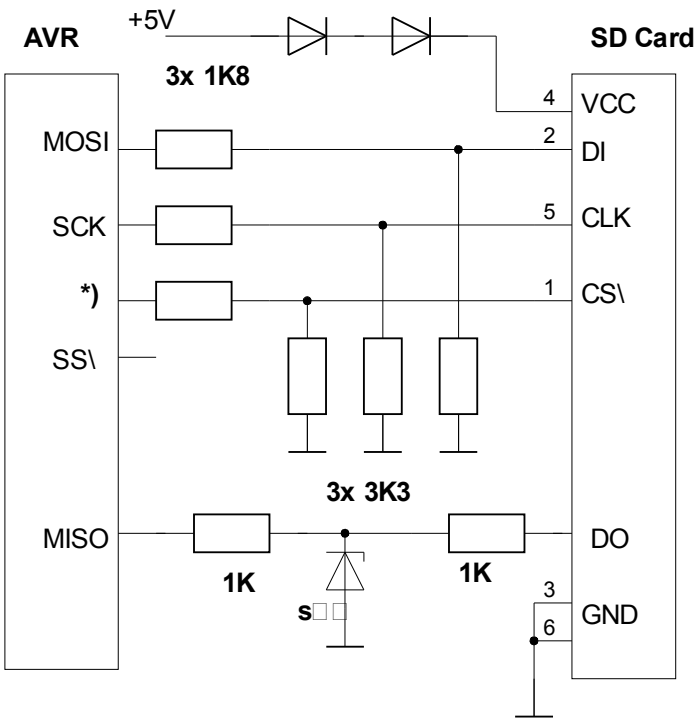
Quellen:

http://members.aon.at/voegel/index.html (AVR-DOS)
http://www.ulrichradig.de/
http://www.roland-riegel.de/sd-reader/
http://elm-chan.org/docs/mmc/mmc_e.html




| | MMC | SD |
|---|---|---|
| 1 | Chip Select(CS) | Chip Select(CS) |
| 2 | CMD/DI | CMD/DI |
| 3 | GND | GND |
| 4 | VCC | VCC |
| 5 | CLK/SCLK | CLK/SCLK |
| 6 | GND | GND |
| 7 | DAT/DO | DAT/DO |
| 8 | NA | NC |
| 9 | NA | NC |

Master - Slave



Die MMC/SD Karte hat einen Spannungsbereich von 2,9 - 3,6V. Die Spannungsversorgung der MMC/SD - Karte wird mit Hilfe von 2 Dioden (je 0,7V Diffusionsspannung) gewonnen.
(Besser ist natürlich ein Stabi!)

Im Prinzip könnte der DO - Anschluss direkt mit dem AVR - Pin verbunden werden, wenn dieser auf Eingang ohne Pullup - Widerstand geschaltet ist. Wenn allerdings der Programmieradapter gleichzeitig mit der Speicherkarte am AVR hängt, könnte es zu 5V-Pegel an DO kommen und die Karte könnte beschädigt werden. Um diese Probleme zu vermeiden, wird eine 3.3V-Zdiode mit 2 Widerständen eingebaut.

Der *) - Anschluss ist der Chip Select. Er kann vomm SS\-Anschluss des SPI kommen oder von einem beliebigen Pin. Allerdings ist in diesem Fall der SS\-Pin nicht beliebig verfügbar, er muss auf 1 liegen da sonst der AVR in den SPI Slave Modus übergeht.

Auf der Homepage von AVR-DOS http://members.aon.at/voegel/index.html gibt es eine BASCOM-Library MMC.LIB, welche es erlaubt, SD-Karten hardwarenah anzusteuern (Lesen und Schreiben von 512 Byte = 1 Sektor).

**<u>Achtung: je nach AVR-Typ liegen die SPI-Portpins anders!</u>**

Zu MMC.LIB gibt es eine Konfigurationsdatei, diese muss natürlich an den verwendeten Mikrokontroller angepassst werden.

Weiter muss für Lesen und Schreiben ein 512 byte Buffer reserviert werden.
Dieser Buffer wird über einen Pointer angesprochen
(Pointer mit VarPtr (buffer(1) erzeugen).

Beispiel in BASCOM:

```
$regfile = "m8def.dat"
$crystal = 3686400                          'Quarz: 3,6864 MHz
$baud = 9600

Dim Abuffer(512) As Byte                    ' Hold Sector to and from CF-Card
Dim Wsrampointer As Word                    ' Address-Pointer for write

'-----------------------------------------------------------------------------
'                     Config_MMC.BAS
'            Config File for MMC Flash Cards Driver
'       (c) 2003-2005 , MCS Electronics / Vögel Franz Josef
'-----------------------------------------------------------------------------
' Place MMC.LIB in the LIB-Path of BASCOM-AVR installation
''....
'' --------- Start of Section for HW-SPI ----------------------------------------

  ' define Chip-Select Pin
  Config Pinb.2 = Output                    ' define here Pin for CS of MMC/SD Card
  Mmc_cs Alias Portb.2
  Set Mmc_cs

  ' Define here SS Pin of HW-SPI of the CPU (f.e. Pinb.0 on M128)
  ' If an other Pin than SS is used for MMC_SS, SS must be set to OUTPUT and high for proper work of SPI
  ' otherwise AVR starts SPI-SLAVE if SS-Pin is INPUT and goes to LOW
  Config Pinb.2 = Output                    ' define here Pin of SPI SS
  Spi_ss Alias Portb.2
  Set Spi_ss                                ' Set SPI-SS to Output and High por Proper work of
                                            ' SPI as Master

  ' HW-SPI is configured to highest Speed
  Config Spi = Hard , Interrupt = Off , Data Order = Msb , Master = Yes , Polarity = High , Phase = 1 , Clockrate = 4 , Noss = 1
'   'Spsr = 1                               ' Double speed on ATMega128
```

```
  Spiinit                              ' Init SPI

' --------- End of Section for HW-SPI -----------------------------------------
' ========== End of user definable range ======================================


' 'Error
Const Cperrdrivereset = 225                  ' Error response Byte at Reset command
Const Cperrdriveinit = 226                  ' Error response Byte at Init Command
Const Cperrdrivereadcommand = 227               ' Error response Byte at Read Command
Const Cperrdrivewritecommand = 228              ' Error response Byte at Write Command
Const Cperrdrivereadresponse = 229              ' No Data response Byte from MMC at Read
Const Cperrdrivewriteresponse = 230             ' No Data response Byte from MMC at Write
Const Cperrdrive = 231
Const Cperrdrivenotsupported = 232              ' return code for DriveGetIdentity, not supported yet

Waitms 1                             ' Wait some time before initialising MMC/SD
Dim Gbdriveerror As Byte                     ' General Driver Error register
Dim Gbdriveerrorreg As Byte                  ' Driver load Error-Register of HD in case of error
Dim Gbdrivestatusreg As Byte                 ' Driver load Status-Register of HD on case of error
Dim Gbdrivedebug As Byte
$lib "MMC.LIB"                       ' link driver library
$external _mmc


Dim I As Word
Dim Sectornr As Long
Wsrampointer = Varptr(abuffer(1))


  Input "Start?" , I
  Gbdriveerror = Driveinit()                 ' Init MMC/SD Card
  Print "INIT  ";
  Print Gbdriveerror


Do
  Input "Sector: " , Sectornr

  Print "Write sector:"
  For I = 1 To 512
    Abuffer(i) = I And &HFF
    Print Abuffer(i);
    Print "  ";
  Next I

  Gbdriveerror = Drivewritesector(wsrampointer , Sectornr )      'dauert 25ms

  Input "Read sector" , Sectornr
  Gbdriveerror = Drivereadsector(wsrampointer , Sectornr )      'dauert 8ms
  Print "Read  ";
  Print Gbdriveerror

  For I = 1 To 512
    Print Abuffer(i);
    Print "  ";
  Next I
   Waitms 1000

Loop
```