Interface the DiosPro to a MMC or SD memory card.

By Michael Simpson

I have created many projects where the ability to store large amounts of data would greatly improve the project. With the price of SDmemory cards so cheap I felt is was time to build an interface for the Dios Pro chip.

You can pickup a 128Meg SD memory cord for less then $10. Ihave seen them new for as little as $7 a few times. If you pick up a card you should shoot for a 32meg to 1Gig card as it will needed to be formatted with Fat16 in order to work with the routines in this application note.

A SD card has a couple of ways you can interface. We will be using the SPI interface as it is simple and sues only 7 of the 9 pins as shown in Figure 1.

The DiosPro has a hardware SPI interface built-in but it only runs at about 600K bps. The routines Ihave provided in the MMC library use a software interface and can reach upwards of 1.2M bps.

There are a couple hurdles to breach before you can successfully interface to one of these cards.

- **3.3V Interface**
- **Physical connection**
- **Low level protocol**
- **FAT16 interface**

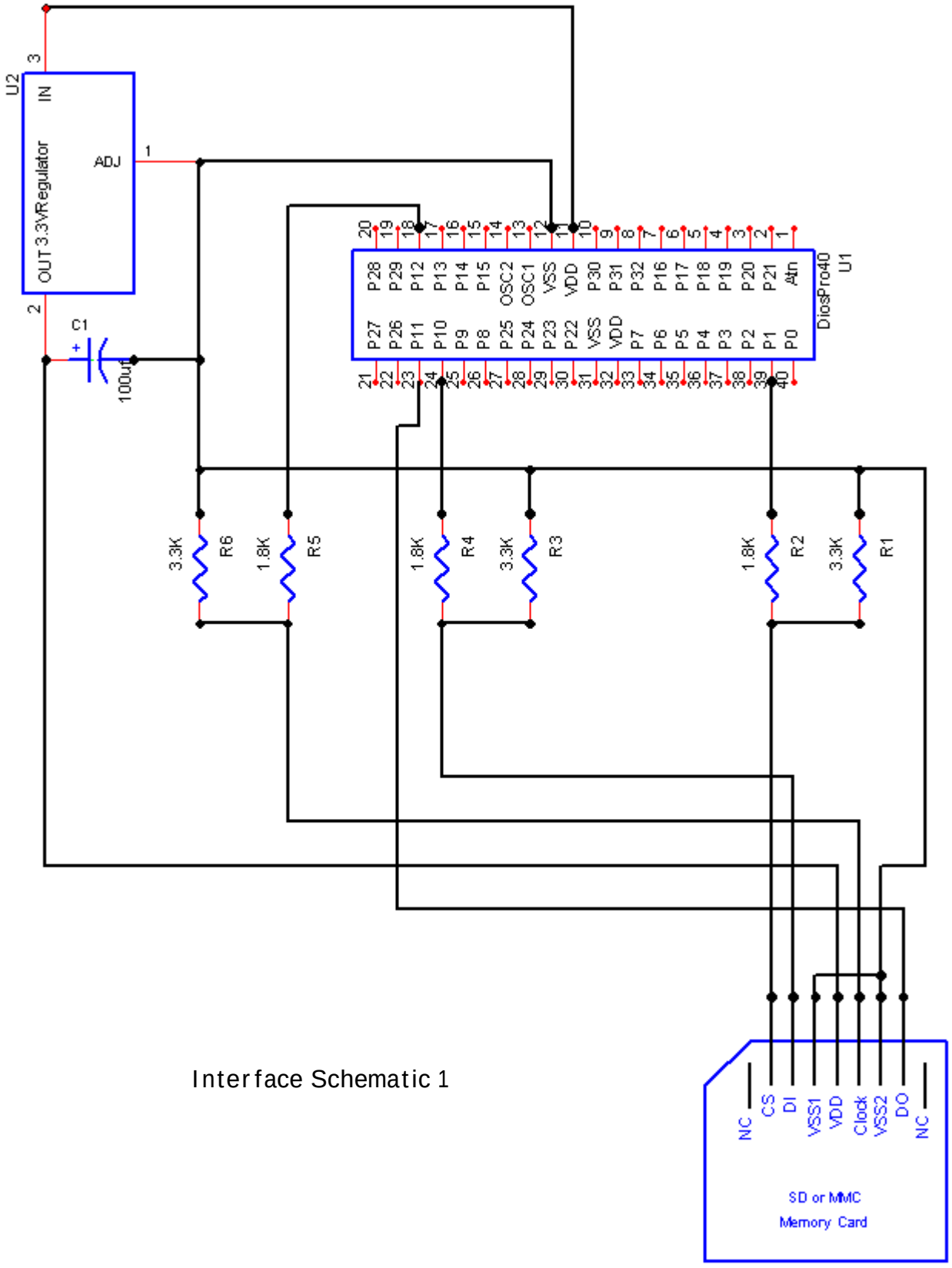
I will address as this application note progresses.

!!! Warning !!!
Do not experiment with the routines on a memory card with critcal data.

3.3 Interface

Lets start with the electrical connection.
The connection only requires 6 resistors and 1 capacitor and a 3.3v regulator to operate.

Kronos Robotics and Electronics
www.KronosRobotics.com



Interface Schematic 1

Take a look at Schematic 1. You will need 3, 1.8K and 3, 3.3K resistors. These create a voltage divider to convert the Dios 5v signals to 3v on the memory card. Going the other way from the memory card to the Dios no level conversion is needed. You will also need a 3.3v regulator. The one I am using costs 65 cents each. I will list all the components at the end of the article. I used a 100uf cap on the regulator. But found just about any value will work.

The IO ports used on the DiosPro are all hard wired except the port used for the CS pin. You must provide this port in the MMCinitfat16 routine.

Please notice that the schematic does not show the support componets that are needed like the resonator or EZ232 driver for programming the chip. These componets are available and included on many of the various Dios carrier boards

Physical Connection

You have a couple ways to connect the card to the DiosPro. You can solder the wires directly to the pads on the card. If you do this you wont be able to plug the card into a PC for data retrieval. Something else that worked very well is an old edge connector like that used on a floppy.

Figure 2

Figure 3

The connector shown in Figures 2 and 3 is a slightly smaller edge connector I had lying around. I cut the cable off and added a header to one end. I used a voltmeter to check the wires to see which was connected to the pins on the card. The floppy edge connector seems to be the most common but they are a bit large. Keep in mind that if you use one of these you can not hot swap the card as the power connectors for hot swap cards need to come in contact first.

Another option is to surf the net and locate a source for SD/MMC memory card socket like the one shown in Figure 4. I have seen the for as little as $4 and as much as $20.

The last option is to purchase an old SD card reader and gut the socket. The one shown in Figure 5 was gutted from a reader that Ipurchased on the web for $5.

Once the connector is chosen you can breadboard it or wire it permanently. Figure 6 shows my bread boarded interface I used to test the various cards and routines. Here Iused a Dios Workboard Deluxe.

Figure 4

Ihave also done a small article that shows how to create a very cool and functional SD connector shown in Figure 8.

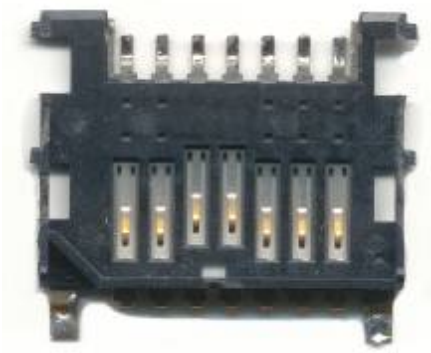


Figure 5

Figure 7

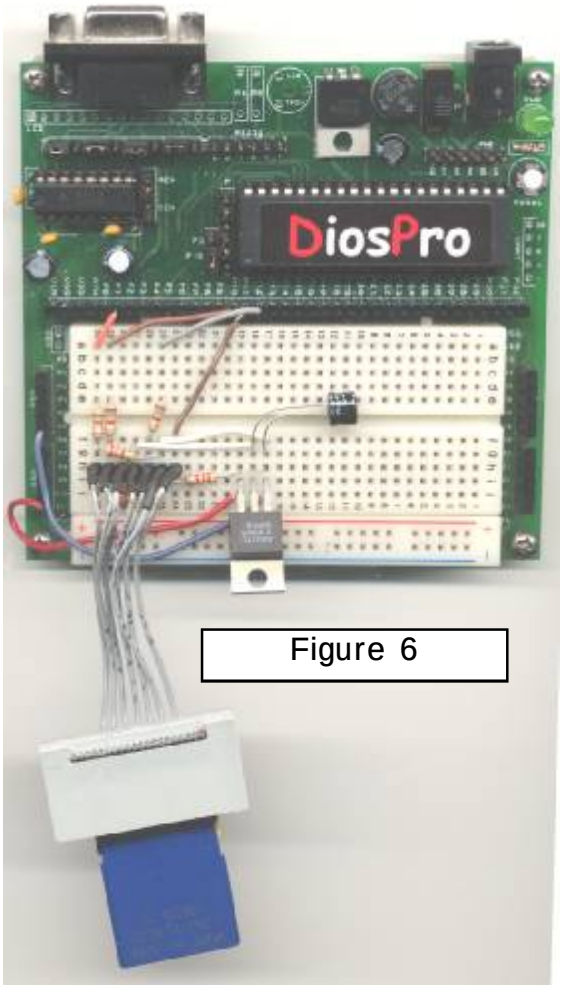


Figure 6

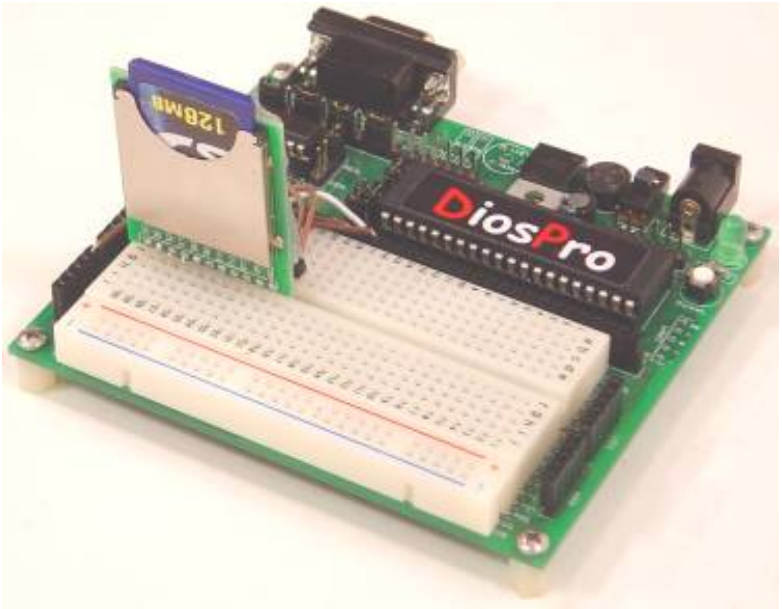


Figure 8

Low Level Routines

You wont have to worry about the low level routines as they are included with the MMC16.lib. This library in included with version 1.4.2 or later compilers. All the routines need are added to your source automatically when you issue the MMCinitfat16() function. Please note that currently only 1 memory card may be connected at a time. Please note that the MMC library is only compatible with the DiosPro28 and DiosPro40 chips.

FAT16 Interface

FAT16 has been implemented in the library but with some limitations for performance reasons. Modern computers have the ability to cache the FAT. We don't have that option so if we were to use FAT16 in the normal fashion we would constantly be reading and writing sectors to various locations. This would make the routines much too slow and use too much memory.

However all is not lost. The main goal is to be able to read and write a file that can be accessed from a PC. The library routine called MMCcreatefile allows you to create a FAT16 file in the root directory. You tell the routine how many clusters you want to pre allocate to this file. What this does is create a file on the memory card on contiguous sectors. Later we can call the MMCopenfile function and it will tell us the sector where the file starts.

Once we know we know the starting location of the file we can use the MMCloadsector and MMCwritesector commands to read and write the raw sectors. A sector on a memory card is 512 bytes. This gives us total random access to the the file. Using a integer variable we can access up to 65535 sectors. That's over 33 million bytes per file. Using other methods we can access all the sectors up to 1Gig.

FAT16 restrictions

- Only Fat16 format is supported
- Only single partition format is supported
- Only 8.3 File names are supported
- Only Root directory is supported with up to 512 files
- Files must be pre-alocated with MMCcreatefile command

Code Examples

dir

This program will display a list of files in the root directory. The start sector and number of sectors is also displayed.

```
DiosPro
'List Files on MMC card
func main()

  dim stat as integer
  stat = MMCinitfat16(1,1)

  if stat = 0 then
    print "Init Error"
    end
  endif

  MMCdir()

endfunc

include \lib\MMC16.lib
```

quickformat

This program will display the files in the root directory then pause for 10 seconds then format the card. This cleans the FAT and clears the root directory. It does not erase the actual data.

```
DiosPro
'MMC quickformat example
'
' !!!! Warning this program will format your
MMC card !!!
func main()

  dim stat as integer
  dim x as integer
  dim startsector(2) as integer
  stat = MMCinitfat16(1,1)

  if stat = 0 then
    print "Init Error"
    end
  endif
  MMCdir()

  for x = 10 to 1 step -1
    print {} "Format in ",x," seconds"
    pause 1000
  next

  print "Formating card do not remove"

  MMCquickformat(1)



endfunc

include \lib\MMC16.lib
```

writefile

This program attempts to open a file. If it can not it will create the file with 500 sectors. It will then write some data to the first 100 sectors.

```
DiosPro
'MMC write file example
func main()
  dim stat as integer
  dim x as integer
  dim sec as integer
  dim filestart(2) as integer
  dim filework(2) as integer

  stat = MMCinitfat16(1,1)
  if stat = 0 then
    print "Init Error"
    end
  endif

  stat=MMCopenfile("BigFile.log",@filestart) '<-- Note pointer
  if stat  = 0 then

    print "Creating new file"
    stat=MMCcreatefile(500,"BigFile.log")
    if stat = 0 then
      print "Unable to create file"
      end
    endif
    stat=MMCopenfile("BigFile.log",@filestart) <-- Note pointer
    if stat  = 0 then
      print "Unable to Open File"
      end
    endif
  endif

'At this point we have the start sector of the file


  'lets write to the first 10 sectors
  print "Write out 5100 bytes

  for sec = 0 to 9
    for x = 0 to 511
      MMCsectorwritebyteA x,sec
    next

   print "write sector ",sec
   MMCaddtoitem filestart,filework,sec
   MMCwritesectorA filework

  next

  print "All done"

endfunc

include \lib\MMC16.lib
```

readfile

This program opens a file then reads the first 100 sectors and diplays them in hex format.

```
DiosPro
'MMC read file example
func main()
  dim stat as integer
  dim x as integer
  dim sec as integer
  dim filestart(2) as integer
  dim filework(2) as integer

  stat = MMCinitfat16(1,1)
  if stat = 0 then
    print "Init Error"
    end
  endif

  stat=MMCopenfile("BigFile.log",@filestart) '<-- Note pointer
  if stat  = 0 then
      print "Unable to Open File"
      end
  endif

'At this point we have the start sector of the file

'lets read to the first 10 sectors
  print "Read in 5100 bytes

  for sec = 0 to 9
    print "read sector ",sec
    MMCaddtoitem filestart,filework,sec
    MMCloadsectorA filework

    MMCdisplaysectorA()
    print
  next


  print "All done"


endfunc

include \lib\MMC16.lib
```

Parts

3, 1.8K resistors
3, 3.3K resistors
1, 100uf capacitor
1, 3.3v regulator . . . . . .jameco 242144

DiosPro40 . . . . . .http://kronosrobotics.com/xcart/customer/product.php?productid=16428
Dios WorkBoard Deluxe . . . . . .http://kronosrobotics.com/xcart/customer/product.php?productid=16452
Dios WorkBoard Basic . . . . . .http://kronosrobotics.com/xcart/customer/product.php?productid=16453

Free Dios Compiler . . . . . .http://www.kronosrobotics.com/downloads/DiosSetup.exe

Other Articles

MMC Library Docs . . . . . http://www.kronosrobotics.com/Projects/MMClib.pdf
MMC 5v Card Interface . . . http://www.kronosrobotics.com/Projects/MMCcon.pdf