

The code has been verified on DE2 board. The top-level module statements is listed in the following. Please note the data width of this code is 1 bit, which is different from the 4 bit of DE4 demo code. The SD mode is used here.

The physical pins SD_CMD , SD_CLK , SD_DAT are all connected to PIO (you can use PIO core in SOPC).

```
// the_SD_CLK
        .out_port_from_the_SD_CLK(SD_CLK),

        // the_SD_CMD
        .bidir_port_to_and_from_the_SD_CMD(SD_CMD),

        // the_SD_DAT
        .bidir_port_to_and_from_the_SD_DAT(SD_DAT),
```

Set the rest of the signals as constants

```
// Set SD Card to SD Mode
assign SD_DAT3 = 1'b1;
```

If you are porting it to DE4 board, please pay attention to the setting of the write protection signal.

The top level software code uses the ucos II operating system which came from NIOS IDE.

//Initialization completes

FS_FILE *myfile;

FS_FILE *myfile1;

FS_Init();

myfile = FS_FOpen("sd_data.txt","w");

myfile1 = FS_FOpen("diag.txt","w");

void task2(void* pdata)

```
{
    int x,i=0;
    //Data to be written to the SD card ptxt
    for(x=0x0;x<0x100;x++)
    {
        //txt[x]=x;
        sprintf(&ptxt[i],"%x",a[x]); // Write formatted data to string
        ptxt[i+1][2]=' ';
        printf("%s\n",ptxt[i]);
        i=i+2;
    }
}
```

```

}

/* create file */

if(save==1)
{
    /* write to file */

    FS_FSeek(myfile,0,FS_SEEK_CUR);
    x = FS_FWrite(ptxt,2,0x200,myfile);
    printf("x=%d",x);

}
else {
    sprintf(mybuffer,"Unable to create file %s\n","sd.txt");
    //_error(mybuffer);
}
GUI_Delay(30);
}
FS_FClose(myfile);
}

if(save==2)
{for (i = 0; i <= 100; i++) {
    PROGBAR_SetValue(ahProgBar, i);

    GUI_Delay(5);
}
    WM_HideWindow(ahProgBar);
    FS_FWrite(telnetdata,1,strlen(telnetdata),myfile1);

    FS_FClose(myfile1);}
//for(i=0;i<320*4;i+=4)
// IOWR_32DIRECT(0X0ff0000, i, 0x11111111);
    OSTimeDlyHMSM(0, 0, 0, 100);

}
}

```

Note: The file system of SD is “uc FS”, whose software package is loaded with read and write drivers for SD Card. See the archive file: SD_FS.rar