```
/*  ============================================================================
 *   Copyright (c) Texas Instruments Inc 2010
 *
 *   Use of this software is controlled by the terms and conditions found in the
 *   license agreement under which this software has been supplied.
 *  ============================================================================
 */

/** @file ata_ext_func.c
 *
 *  @more ATAFS interface functions
 *
 *
 * ============================================================================
 */

/* ============================================================================
 * Revision History
 * ================
 * 30-Jun-2010 Created
 * ============================================================================
 */

#include "ata.h"

AtaError ATA_fopen(AtaFile *pAtaFile, char *name, char *ext)
{
    char FileName[256];
    char nameUp[10];
    char extUp[10];
    short counter;
    AtaError ata_error;

    ata_error = ATA_ERROR_NONE;

    // check name.ext
    if ((strlen(name)>8)||(strlen(ext)>3)) {
      ata_error = ATA_ERROR_UNSUPPORTED;
      return ata_error;
    }

    for (counter=0; counter<strlen(name); counter++)
    {
        nameUp[counter] = toupper(name[counter]);
    }
    nameUp[counter] = 0;

    for (counter=0; counter<strlen(ext); counter++)
    {
        extUp[counter] = toupper(ext[counter]);
    }
    extUp[counter] = 0;

    //pAtaFile->pDrive = gpstrAtaDrive;

    // find the first file in this drive
    //ata_error =  ATA_fileInit(pstrAtaDrive, pAtaFile);
    ATA_findFirst(pAtaFile);

    // find the file, if it exists
    do
    {
        // get the file name for this entry
        ata_error = ATA_getLongName(pAtaFile, &FileName[0], 0, strlen(name)
+strlen(ext)+1);

        // file name is the same?
        if ((strncmp(FileName,name,strlen(name)) !=
```

```
0)&&(strncmp(FileName,nameUp,strlen(nameUp)) != 0))
        {
            // if the file name is not the same, get the next entry
            ata_error = ATA_findNext(pAtaFile);
            // have we get the empty entry?
            if(ata_error==ATA_ERROR_FILE_NOT_FOUND)
            {
                break;
            }
        }
        else
        {
            // if the file name is the same, compare the extension name
            for(counter=0;counter<strlen(ext);counter++)
            {
                FileName[counter]=FileName[strlen(name)+counter+1];
            }
            if ((strncmp(FileName,ext,strlen(ext)) !=
0)&&(strncmp(FileName,extUp,strlen(extUp)) != 0))
            {
                ata_error = ATA_findNext(pAtaFile);
                if(ata_error==ATA_ERROR_FILE_NOT_FOUND)
                {
                    break;
                }
            }
            else
            {
                // we have a match
                return ata_error;
            }
        }
    } while (1);

    // if the file does not exist, then create it using the empty entry and return the
file descriptor
    if (ata_error==ATA_ERROR_FILE_NOT_FOUND)
    {
        // set the file name and extension name
        ata_error = ATA_setFileName(pAtaFile, name, ext);

        /* Create a file, returns 0 for successful creation*/
        ata_error |= ATA_create(pAtaFile);
    }

    return ata_error;
}

AtaError ATA_delete_file(AtaFile *pAtaFile, char *fileNameIn) {

    AtaError ata_error;
    char     FileName[256];
    char nameUp[10];
    short counter;

    // check fileNameIn
    if (strlen(fileNameIn)>8) {
      ata_error = ATA_ERROR_UNSUPPORTED;
      return ata_error;
    }

    for (counter=0; counter<strlen(fileNameIn); counter++)
    {
        nameUp[counter] = toupper(fileNameIn[counter]);
    }
    nameUp[counter] = 0;

    //pAtaFile->pDrive = gpstrAtaDrive;
```

```
    // find the first file in this drive
    //ata_error =  ATA_fileInit(pstrAtaDrive, pAtaFile);
    ATA_findFirst(pAtaFile);

    do
    {
        // get the file name for this entry
        ATA_getLongName(pAtaFile, &FileName[0], 0, 20);
        if(strcmp(FileName,nameUp) == 0)
        {
            ata_error = ATA_delete(pAtaFile);
            if (ata_error == 0)
            {
                printf("\nATA_delete of %s successful\n",FileName);
            } else
            {
                printf("\nATA_delete of %s unsuccessful\n",FileName);
            }
            return(ata_error);
        } else
        {
            // find next file
            ata_error=ATA_findNext(pAtaFile);
            if (ata_error==ATA_ERROR_FILE_NOT_FOUND)
            {
                printf("\nfile not found\n");
                return(ata_error);
            }
        }
    } while (1);
}

AtaError CD(AtaFile *pAtaDir, char *dirNameIn)
{
    AtaError ata_error;
    char dirName[30];
    char nameUp[10];
    short counter;

    // check dirNameIn
    if (strlen(dirNameIn)>8) {
      ata_error = ATA_ERROR_UNSUPPORTED;
      return ata_error;
    }

    for (counter=0; counter<strlen(dirNameIn); counter++)
    {
        nameUp[counter] = toupper(dirNameIn[counter]);
    }
    nameUp[counter] = 0;

    ata_error=ATA_findFirst(pAtaDir);
    //  printf("ATA_findFirst, error=0x%x\n", ata_error);
    if (ata_error!=ATA_ERROR_NONE) return ata_error;

    ata_error=ATA_getLongName(pAtaDir, &dirName[0], 0, 20);
    //  printf("ATA_findFirst, error=0x%x, name=%s\n", ata_error, dirName);
    if (ata_error!=ATA_ERROR_NONE) return ata_error;

    while (strcmp(dirName,nameUp)!=0)
    {
        ata_error=ATA_findNext(pAtaDir);
        //    printf("ATA_findNext, error=0x%x\n", ata_error);
        if (ata_error!=ATA_ERROR_NONE) return ata_error;
        ata_error=ATA_getLongName(pAtaDir, &dirName[0], 0, 20);
        //    printf("ATA_findNext(loop), error=0x%x, name=%s\n", ata_error, dirName);
        if (ata_error!=ATA_ERROR_NONE) return ata_error;
    }
```

```
    ata_error=ATA_cd(pAtaDir);
    //  printf("ATA_cd, error=0x%x\n", ata_error);
    if (ata_error!=ATA_ERROR_NONE) return ata_error;

    return ata_error;
}
```