文件操作例程

1.程序功能

实现在控制器目录内文件写入、文件读取、文件拷贝、文件删除、文件重命名等功能。

控制器LC3000根目录文件路径：/mnt/data/app。

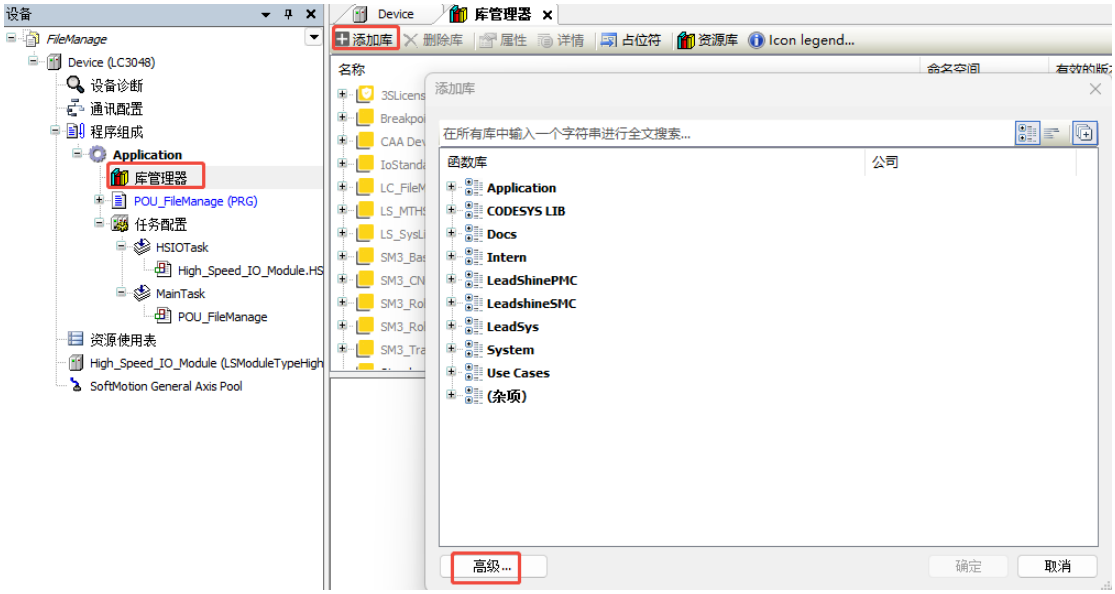
仿真时文件路径：D:/zj/123.doc（D盘文件夹zj里的文件123.doc）

注意事项：

- 一般情况下建议用户将文件放在根目录下（/mnt/data/app），或者用户新建的文件下

2.程序编写步骤

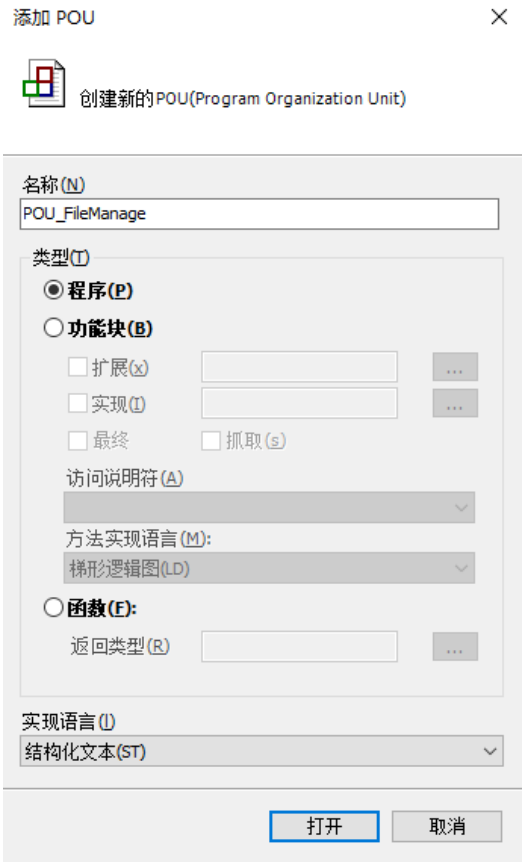
1) 新建工程并命名例FileManage，编程语言为结构化文本ST，然后双击左侧设备栏“库管理器”分别添加“SysFile,3.5.15.0(System)”，“SysTypes Interfaces,3.5.2.0(System)”，“CmpErrors,3.3.1.40(System)”文件管理库，如下图所示；点击高级，输入库名称搜索，选择库进行添加。



图： 添加库文件

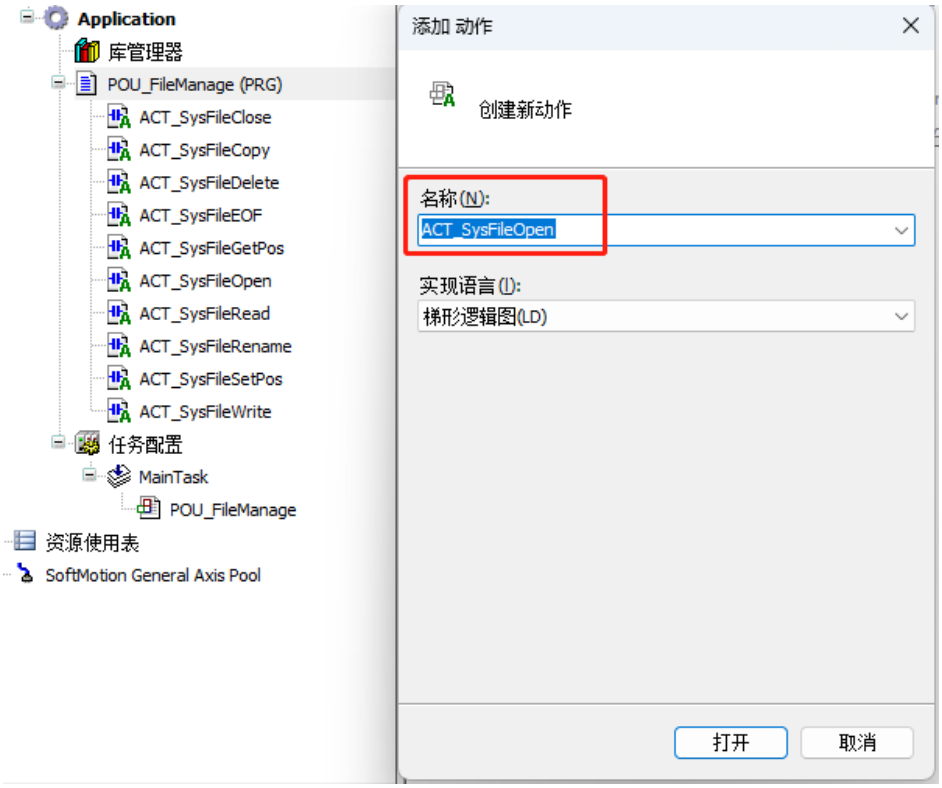
2) 右击左侧设备栏“Application”，选择“添加对象”-“程序组织单元”，命名为

POU_FileManage，选择类型为程序，实现语言为结构化文件，点击打开即可，如下图所示。



图：添加 POU

3) 右键“POU_FileManage”选择“添加对象”–“动作”，命名为ACT_SysFileOpen，选择实现语言为梯形图LD，点击打开即可，如下图所示；相同的操作添加ACT_SysFileClose等其他动作。



图：添加动作

4) 在第三步添加的动作中完成各模块程序功能的编写，如下图1所示；在主程序POU_FileManage中控制文件操作使能变量状态来实现文件具体操作，并加入相关IO操作，如下图3所示；其中定义的几个比较重要的变量如下图2所示。

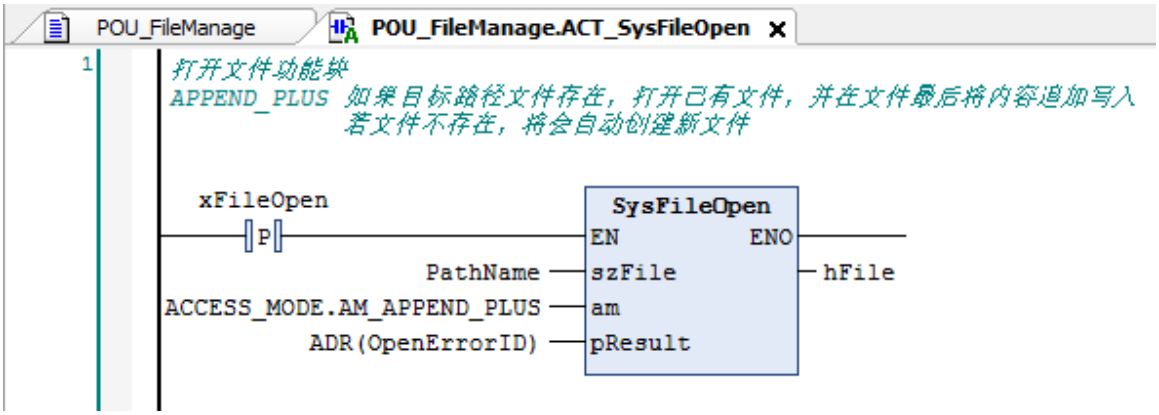


图 1：打开文件功能块

```
//ACT_SysFileOpen 变量
    xFileOpen: BOOL := FALSE;  //打开文件使能
    PathName: STRING := '/home/userftp/testl.txt';  //文件路径及文件名
    OpenErrorID: RTS_IEC_RESULT;   //文件打开错误码
    hFile: RTS_IEC_HANDLE := RTS_INVALID_HANDLE;  //文件句柄
//ACT_SysFileClose 变量
    xFileClose: BOOL := FALSE;   //关闭文件使能
    FileCloseErrorID: RTS_IEC_RESULT;  //文件关闭错误码
//ACT_SysFileCopy 变量
    xFileCopy: BOOL := FALSE; //复制文件使能
    CoDestFileName: STRING :='/home/userftp/testl_copy.txt' ; //目标文件路径及文件名
    CoSourceFileName: STRING := '/home/userftp/testl.txt';  //源文件路径及文件名
    CoSize: __XWORD := 0;  //实际复制文件大小，单位字节
    FileCopyErrorID: RTS_IEC_RESULT :=0;   //文件复制错误码
//ACT_SysFileDelete 变量
    xFileDelete: BOOL := FALSE;  //删除文件使能
    DelFileName: STRING :='/home/userftp/testl.txt';  //删除的文件路径及文件名
    FileDeleteErrorID: RTS_IEC_RESULT;  //文件删除错误码
//ACT_SysFileRead 变量
    xFileRead: BOOL := FALSE;   //读取文件数据使能
    ReadData: STRING(128) ;//读取的数据
    ReadErrorID: RTS_IEC_RESULT;  //文件读取错误码
    RelReadSize: __XWORD := 0;    //实际读取数据大小，单位字节
//ACT_SysFileWrite 变量
    xFileWrite: BOOL := FALSE;   //写入文件数据使能
    SentData: STRING(128) := 'This is the first line.$r'; //写入文件内容（$r为回车字符）
    WriteErrorID: RTS_IEC_RESULT;  //文件写入错误码
    RelSentSize: __XWORD := 0;   //实际写入文件数据大小，单位字节
//ACT_SysFileSetPos 变量
    xFileSetPos: BOOL := FALSE;   //设定文件指针到给定位置
    Offset: __XWORD := 0 ;  //文件指针偏移量
    FileSetPosErrorID: RTS_IEC_RESULT;  //设定文件指针到给定位置错误码
//ACT_SysFileGetPos 变量
    xFileGetPos: BOOL := FALSE;   //获取文件指针的实际位置
    FileGetPosErrorID: RTS_IEC_RESULT;  //获取文件指针的实际位置错误码
    pulPOS: __XWORD := 1;   //文件指针的实际位置
//ACT_SysFileRename 变量
    xFileRename: BOOL := FALSE;  //重命名文件使能
    OldFileName: STRING := '/home/userftp/testl_copy.txt';    //旧文件路径及文件名
    NewFileName: STRING :='/home/userftp/test2.txt';    //新文件路径及文件名
    FileRenameErrorID: RTS_IEC_RESULT;   //文件重命名错误码
//ACT_SysFileEOF 变量
    xFileEOF: BOOL := FALSE;  //判断文件内容是否全部读取使能
    FileEOFErrorID: RTS_IEC_RESULT; //ERR_OK表示文件内容已全部读取
```

图 2：重要的变量

```
//将 iFileState置为1，开始执行
CASE iFileState OF
    0:
      ;
    1: //打开根目录下文件或者创建文件 " test1.txt "
      PathName :='/home/userftp/test1.txt';
      xFileOpen := TRUE;
      IF hFile <> RTS_INVALID_HANDLE    THEN    //文件是否打开判断 [6 lines]
      END_IF

    2://在test1.txt文件内写入数据
      SentData := 'This is the first line.$r';    //写入的数据内容($r为回车字符)
      xFileWrite:= TRUE;
      IF RelSentSize <> 0 THEN      //文件内容是否写入判断 [7 lines]
      END_IF
      ;
    3://设置test1.txt文件指针实际位置
      Offset:=0;
      xFileSetPos:= TRUE;
      xFileGetPos:= TRUE;
      IF pulPOS=0 AND FileSetPosErrorID = 0 AND FileGetPosErrorID = 0 THEN [4 lines]
      ELSIF  FileSetPosErrorID <> 0 OR FileGetPosErrorID <> 0 THEN  [4 lines]
      END_IF
      ;
    4://读取test1.txt文件全部数据内容
      xFileRead:= TRUE;
      xFileEOF:= TRUE;
      IF RelReadSize <> 0 AND FileEOFErrorID= 0 THEN [4 lines]
      ELSIF  ReadErrorID <> 0 OR FileEOFErrorID <> 0 THEN  [4 lines]
      END_IF
      ;
    5://关闭指定文件
      xFileClose:=TRUE;
      TON_0(IN:= xFileClose, PT:= T#100MS, Q=> , ET=> );  //关闭文件延时判断
      IF TON_0.Q THEN [8 lines]
      END_IF
      ;
    6://复制指定文件
      CoDestFileName:='/home/userftp/test1_copy.txt';
      CoSourceFileName :='/home/userftp/test1.txt';
      xFileCopy:=TRUE;
      IF CoSize <> 0 AND FileCopyErrorID = 0 THEN [3 lines]
      ELSIF CoSize = 0 AND FileCopyErrorID <> 0 THEN
            xFileCopy:=FALSE;
            iFileState:=10;
            RETURN;
      END_IF
```

图 3：文件操作主程序

5) 新增加一个任务，用于处理文件的操作。配置步骤：右击左侧设备栏“任务配置”=> 选择“添加对象” => “任务” => 命名为Task_FileManage => 将

POU_FileManage加进扫描区 => 设置任务优先级为31，设置扫描周期为100ms，如下图所示。



图：添加文件处理任务

6) 完成编程，编译OK后连接控制器，将程序下载到控制器中，点击运行按钮或者按F5，运行程序。然后根据功能需求情况相应地强制使能条件为TRUE来调试。

3.运行结果：

1) 打开文件：当打开文件使能被强制为TRUE后，打开（或创建并打开）指定路径下的文件，如下图所示。

输入am有六种模式：

①.READ：如果目标路径文件存在，打开已有文件，不存在，返回错误值。

②.WRITE：如果目标路径文件存在，则对文件重新写入，不存在，将会自动创建新文件，再写入。

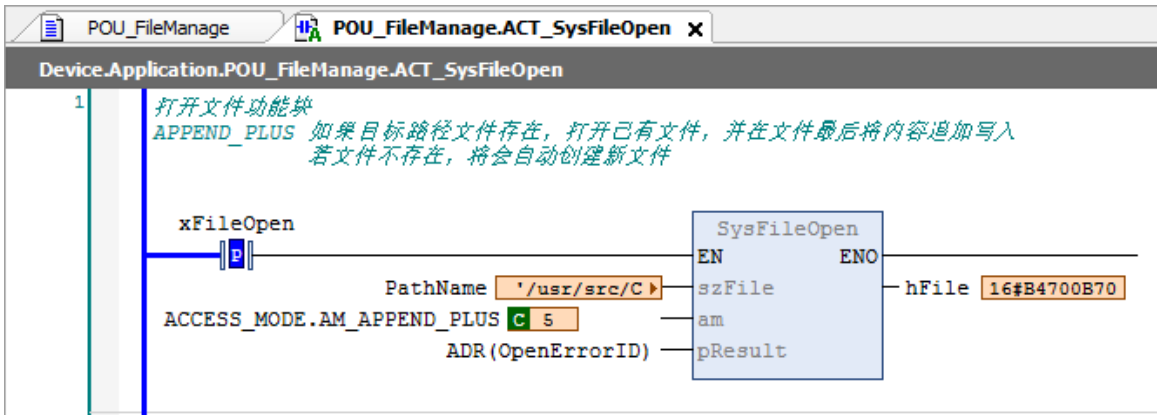
③.APPEND：如果目标路径文件存在，打开已有文件，并再文件最后将内容追加写入，不存在，返回错误值。

④.READ_PLUS：打开文件，并具有读写权限。如果目标路径文件存在，打开已有文件，不存在，返回错误值。

⑤.WRITE_PLUS：如果目标路径文件存在，则对文件重新写入，不存在，将会

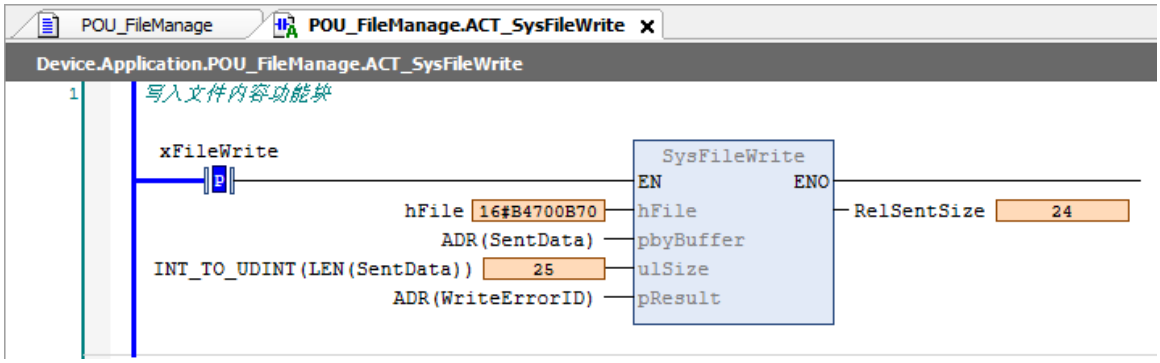
自动创建具有读/写权限新文件，再写入。

⑥.APPEND_PLUS：相比APPEND，APPEND_PLUS更常用，相比前者，后者文件不存在，将会自动创建新文件。



图： 打开文件

2) 写入文件内容：当写入文件内容使能被强制为TRUE后，设置的内容被写入文件句柄hFile对应的文件中，如下图所示。



图：写入文件内容

3) 设置并读取当前文件指针实际位置：当xFileSetPos被强制为TRUE后，将预设偏移值offset设定为文件指针位置，如下图1所示；当xFileGetPos被强制为TRUE后，读取当前文件指针的实际位置，如下图2所示；

注意事项：

- 执行完SysFileWrite指令后，文件指针在末尾位置，必须先将文件指针的位置设定到0或其他，再执行SysFileRead指令读取文件内容；
- 设定文件指针，实际是设置读取数据的偏移值；例如数据“This is the first line.$R”；若偏移值offset为0，则读取完整数据；若offset为10，则从第11个字符开始读取，读取数据内容为“e first line.$R”。
- 设定完文件指针后，可用SysFileGetPos指令读取实际的指针位置。

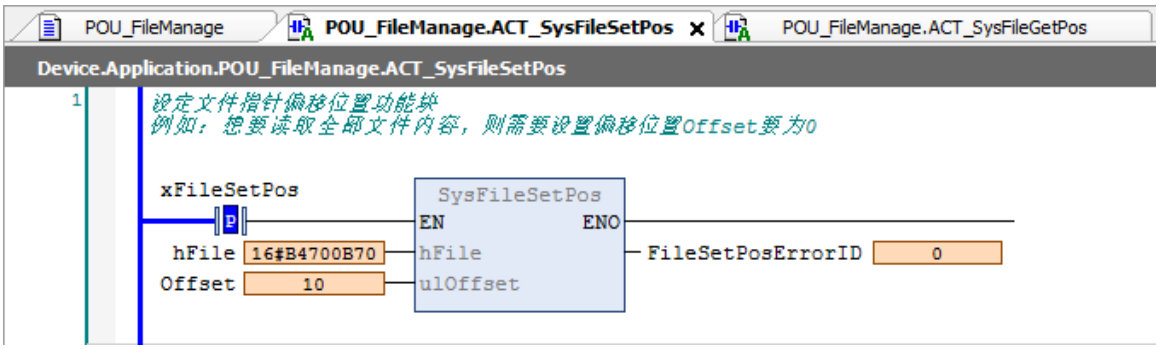


图 1：设定文件指针位置

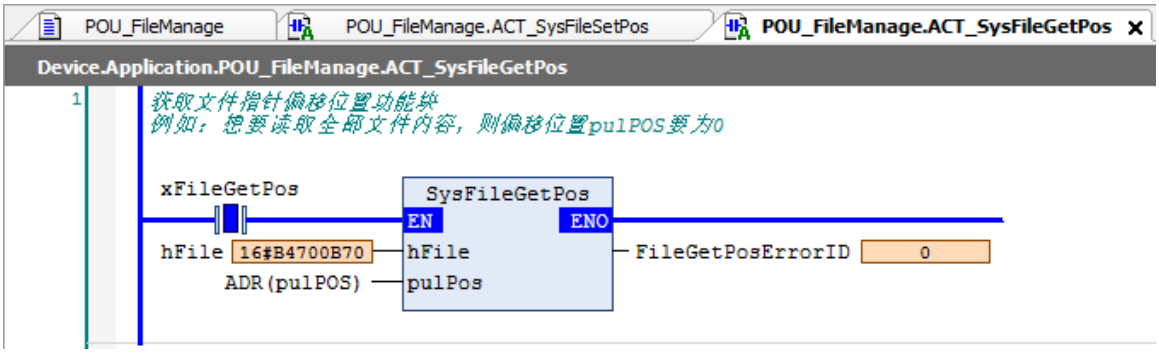
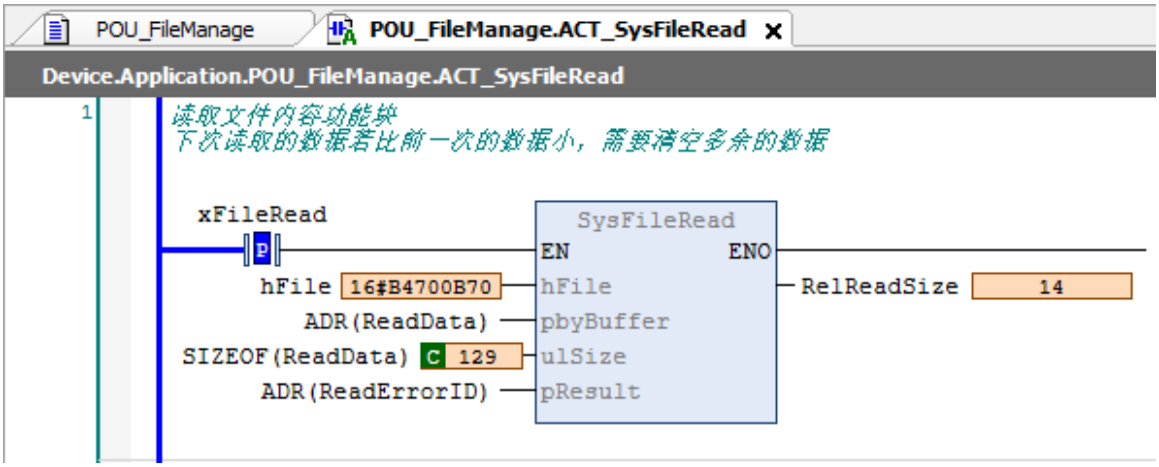


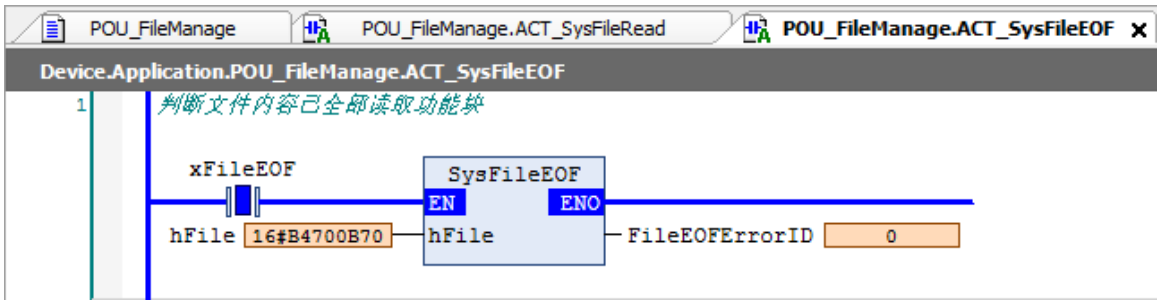
图 2：读取文件当前指针位置

4) 读取文件内容：当读取文件内容使能被强制为TRUE后，读取文件句柄hFile对应的文件内容到ReadData中，如下图所示。



图：读取文件内容

5) 判断文件内容已全部读取：当xFileEOF被强制为TRUE后，读取当前文件内容是否已全部读取，若已读取，则FileEOFErrorID为0；

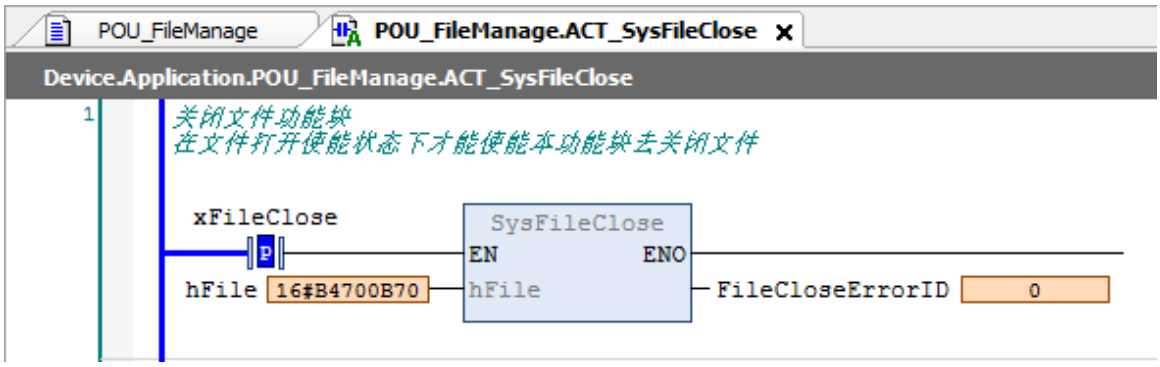


图：判断文件内容已全部读取

6) 关闭文件：当关闭文件使能被强制为TRUE后，关闭文件句柄hFile对应的文件，如下图所示。

注意事项：

- 文件操作完成后必须要关闭文件，否则会导致内存溢出等问题，严重使PLC系统崩溃。
- 执行文件关闭指令后，需要把文件句柄hFile复位为初始值RTS_INVALID_HANDLE。

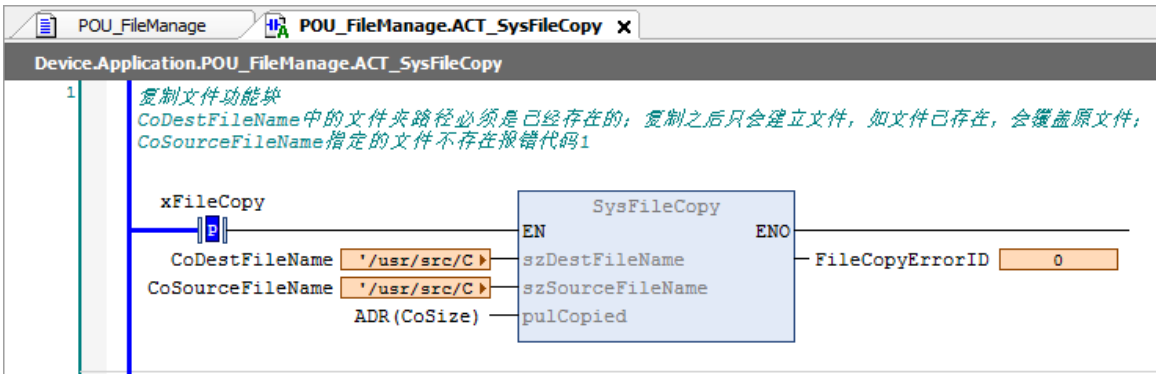


图：关闭文件

7) 拷贝文件：当拷贝文件使能被强制为TRUE后，文件szSourceFileName内容被拷贝到szDestFileName中，如下图所示。

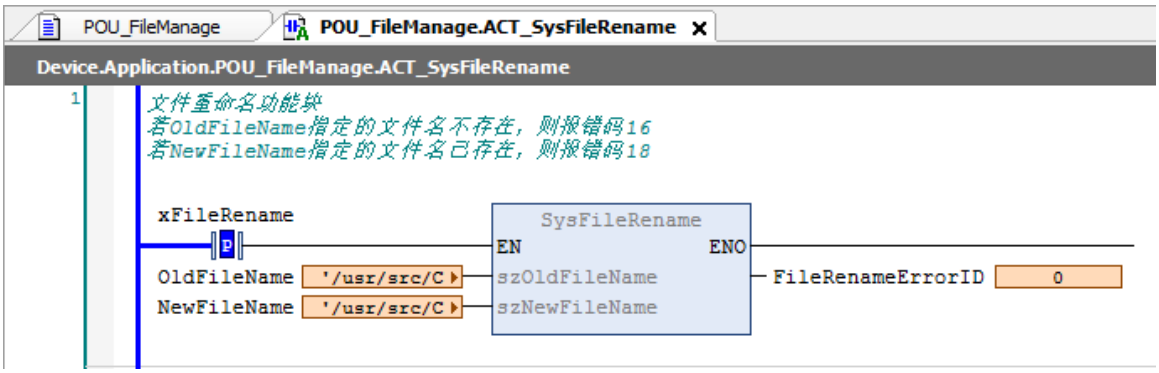
注意事项：

- szDestFileName中的文件夹路径必须是已经存在的；复制之后只会建立文件，如文件已存在，会覆盖原文件；
- szSourceFileName指定的文件不存在报错代码1；



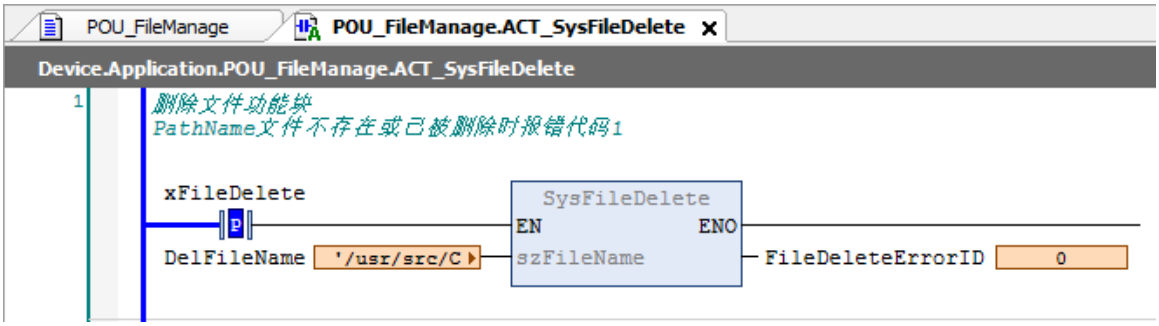
图：拷贝文件

8) 重命名文件：当重命名文件使能被强制为TRUE后，文件名称由szOldFileName被改成 szNewFileName，如下图所示。



图：重命名文件

9) 删除文件：当删除文件使能被强制为 TRUE 后，文件 szFileName 被删除，如下图所示。



图：删除文件

经过上述 1-9 文件操作后，最后在控制器目录下 /mnt/data/app/文件夹内只剩下文件“test2.txt”，如下图 1 所示；将文件拷贝到本地电脑的路径下，在电脑上打开，如下图 2 所示。

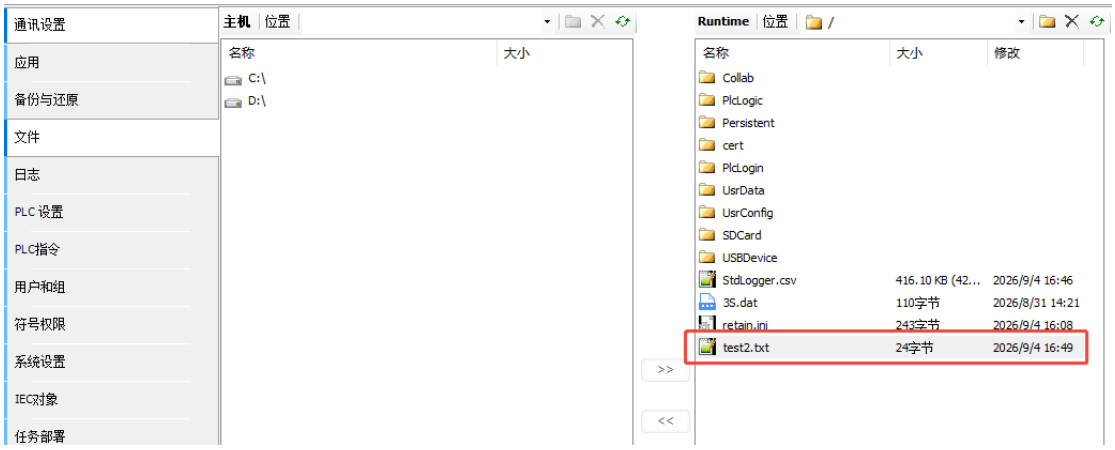


图 1：控制器根目录内文件

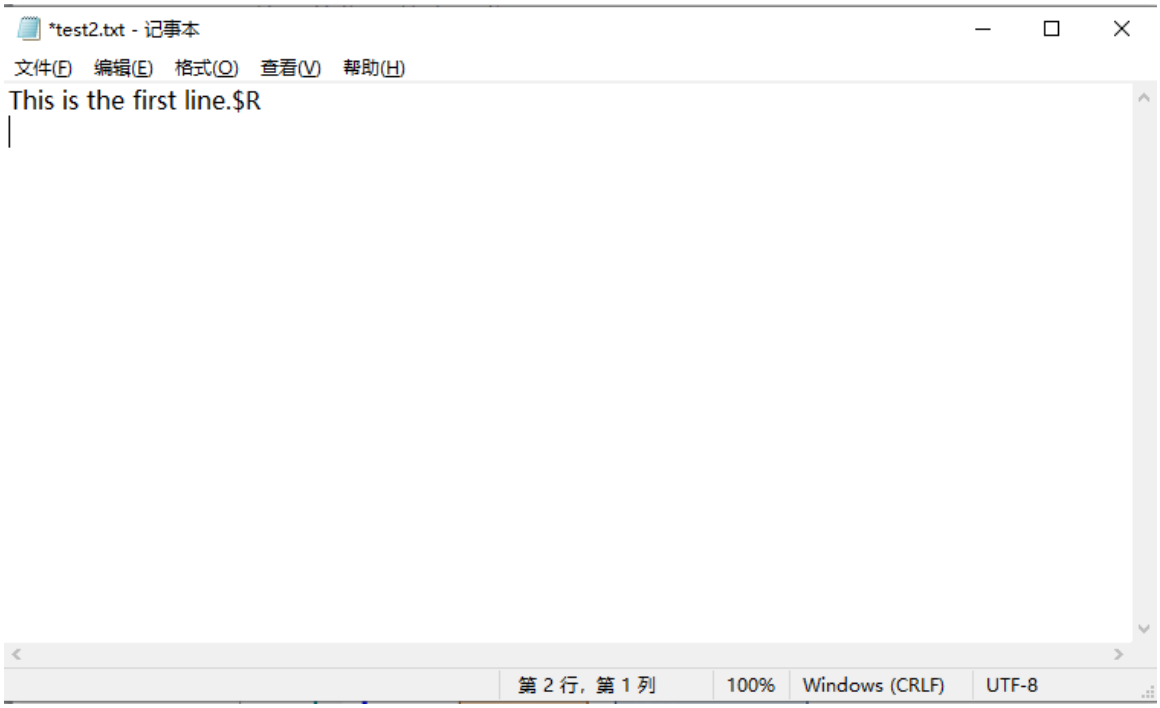


图 2：test2.txt 文件内容

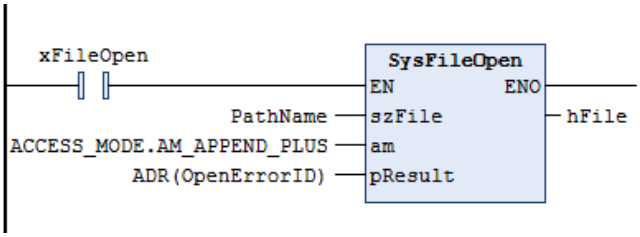
本例程原代码参见 LC3000 软件资料中的“例程”文件夹中的“**文件操作功能**”。

3. 指令介绍

3.1 SysFileOpen

本指令用于打开或创建指定路径下的文件。

- **指令格式**

指令	名称	FB/FC	LD表现	库
SysFileOpen	文件打开	FC		SysFile,3.5.15.0(System)

- **相关变量**

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
szFile	文件名	String	-	-	文件名。文件名可以包含文件的绝对路径或相对路径。路径条目必须用斜杠（/）分隔，不能用反斜杠（\）。
am	文件打开模式	ACCESS_MODE	-	-	READ：如果目标路径文件存在，打开已有文件，不存在，返回错误值。 WRITE：如果目标路径文件存在，则对文件重新写入，不存在，将会自动创建新文件，再写入。 APPEND：如果目标路径文件存在，打开已有文件，并再文件最后将内容追加写入，不存在，返回错误值。 READ_PLUS：打开文件，并具有读写权限。如果目标路径文件存在，打开已有文件，不存在，返回错误值。 WRITE_PLUS：如果目标路径文件存在，则对文件重新写入，不存在，将会自动创建具有读/写权限新文件，再写入。 APPEND_PLUS：相比 APPEND，APPEND_PLUS 更常用，相比前者，后者文件不存在，将会自动创建新文件。

pResult	文件打开错误码指针	POINTER TO RTS_IEC_RESULT	-	-	指向文件打开错误码

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
hFile	返回值	RTS_IEC_HANDLE	-	-	文件句柄，作为后续文件操作功能的输入

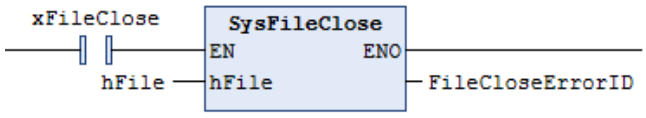
- **功能说明：**

本指令用于打开或创建指定路径下文件，文件路径及文件名由 szFile 指定；文件打开模式由 am 指定；返回文件句柄，作为写文件等功能块的输入；打开文件错误码请查看库 CmpErrors,3.3.1.40(System)。

3.2 SysFileClose

本指令用于文件读写操作结束后关闭文件。

- **指令格式**

指令	名称	FB/FC	LD表现	库
SysFileClose	关闭文件	FC		SysFile,3.5.15.0(System)

- **相关变量**

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
hFile	文件句柄	RTS_IEC_HANDLE	-	-	文件句柄

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述

FileCloseErrorID	返回值	RTS_IEC_RESULT	-	-	关闭文件错误码

- **功能说明：**

本指令用于文件读写操作结束后关闭文件。关闭文件错误码请查看库 CmpErrors,3.3.1.40(System)。文件操作完成后必须要关闭文件，否则会导致内存溢出等问题，严重使 PLC 系统崩溃

3.3 SysFileCopy

本指令用于复制指定文件。

- **指令格式**

指令	名称	FB/FC	LD表现	库
SysFileCopy	复制文件	FC		SysFile,3.5.15.0(System)

- **相关变量**

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
szDestFileName	目标文件	STRING	-	-	目标文件路径及文件名
szSourceFileName	源文件	STRING	-	-	源文件路径及文件名
pulCopied	实际读取数据大小指针	POINTER TO __XWORD	-	-	实际读取数据大小指针；单位：字节

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述

FileCopyErrorID	返回值	RTS_IEC_RESULT	-	-	复制文件错误码

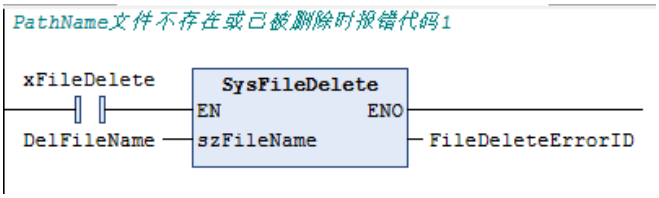
- **功能说明：**

本指令用于复制文件；CoDestFileName 中的文件夹路径必须是已经存在的，复制之后只会建立文件，如文件已存在，会覆盖原文件；CoSourceFileName 指定的文件不存在会报错误代码 1；复制文件错误码请查看库 CmpErrors,3.3.1.40(System)。

3.4 SysFileDelete

本指令用于删除指定文件。

- **指令格式**

指令	名称	FB/FC	LD表现	库
SysFileDelete	删除文件	FC		SysFile,3.5.15.0(System)

- **相关变量**

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
szFileName	指定文件名	STRING	-	-	指定文件路径及文件名

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
FileDeleteErrorID	返回值	RTS_IEC_RESULT	-	-	删除文件错误码

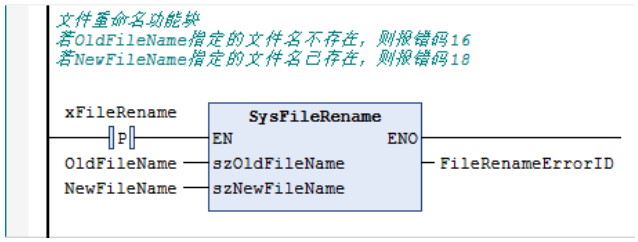
- **功能说明：**

本指令用于删除指定文件。删除文件错误码请查看库 CmpErrors,3.3.1.40(System)。

3.5 SysFileRename

本指令用于重命名指定文件。

- **指令格式**

指令	名称	FB/FC	LD表现	库
SysFileRename	重命名文件	FC		SysFile,3.5.15.0(System)

- **相关变量**

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
szOldFileName	旧文件路径及文件名	STRING	-	-	旧文件路径及文件名
sz NewFileName	新文件路径及文件名	STRING	-	-	新文件路径及文件名

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
FileRename ErrorID	返回值	RTS_IEC_RESULT	-	-	重命名文件错误码

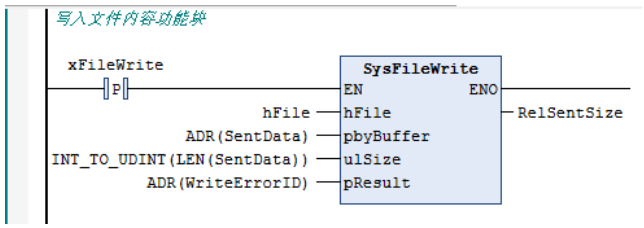
- **功能说明：**

本指令用于重命名指定文件；若 OldFileName 指定的文件名不存在，则报错码 16；若 NewFileName 指定的文件名已存在，则报错码 18。重命名文件错误码请查看库 CmpErrors,3.3.1.40(System)。

3.6 SysFileWrite

本指令用于在文件打开后，写入文件内容。

- **指令格式**

指令	名称	FB/FC	LD表现	库
SysFileWrite	写入文件内容	FC		SysFile,3.5.15.0(System)

- **相关变量**

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
hFile	文件句柄	RTS_IEC_HANDLE	-	-	文件句柄
pbyBuffer	要写入的文件内容指针	POINTER TO STRING	-	-	要写入的文件内容指针
ulSize	要写入内容的字节数	__XWORD	-	-	要写入内容的字节数
pResult	写入文件内容错误码指针	POINTER TO RTS_IEC_RESULT	-	-	写入文件内容错误码指针

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
RelSentSize	返回值	__XWORD	-	-	实际写入文件数据大小，单位字节

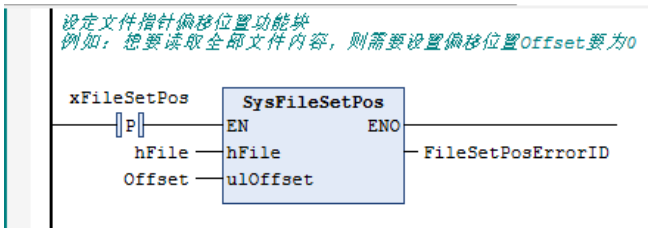
- **功能说明：**

本指令用于在文件打开后，写入文件内容。写入文件内容错误码请查看库CmpErrors,3.3.1.40(System)。

3.7 SysFileSetPos

本指令用于设置文件指针到指定位置。

- **指令格式**

指令	名称	FB/FC	LD表现	库
SysFileSetPos	设置文件指针到指定位置	FC		SysFile,3.5.15.0(System)

- **相关变量**

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
hFile	文件句柄	RTS_IEC_HANDLE	-	-	文件句柄
uloffset	偏移位置	__XWORD	-	-	文件指针偏移量

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
FileSetPosErrorID	返回值	RTS_IEC_RESULT	-	-	设置文件指针到指定位置错误码

- **功能说明：**

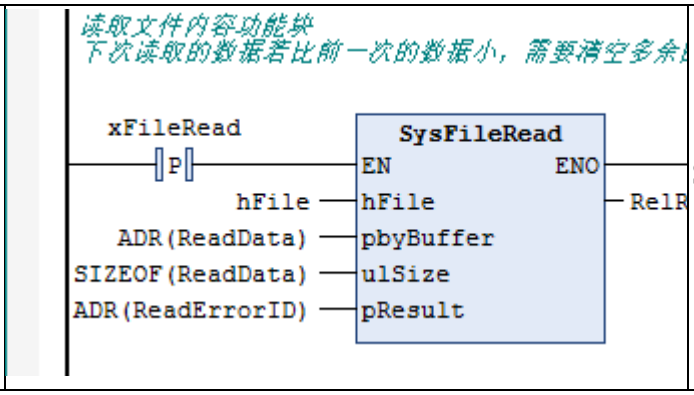
本指令用于设定文件指针到指定位置。设定文件指针到指定位置错误码请查看库CmpErrors,3.3.1.40(System)。

3.8 SysFileRead

本指令用于在文件打开后，读取文件的数据内容。

- **指令格式**

指令	名称	FB/FC	LD表现	库

SysFileRead	读取文件数据内容	FC		SysFile,3.5.15.0(System)

- **相关变量**

输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
hFile	文件句柄	RTS_IEC_HANDLE	-	-	文件句柄
pbyBuffer	读取到的文件内容指针	POINTER TO STRING	-	-	读取到的文件内容指针
ulSize	要读取文件的字节数	__XWORD	-	-	要读取文件的字节数
pResult	读取文件内容错误码指针	POINTER TO RTS_IEC_RESULT	-	-	读取文件内容错误码指针

输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
RelSentSize	返回值	__XWORD	-	-	实际读取文件数据大小，单位字节

- **功能说明：**

本指令用于在文件打开后，读取文件的数据内容。