**文件处理**

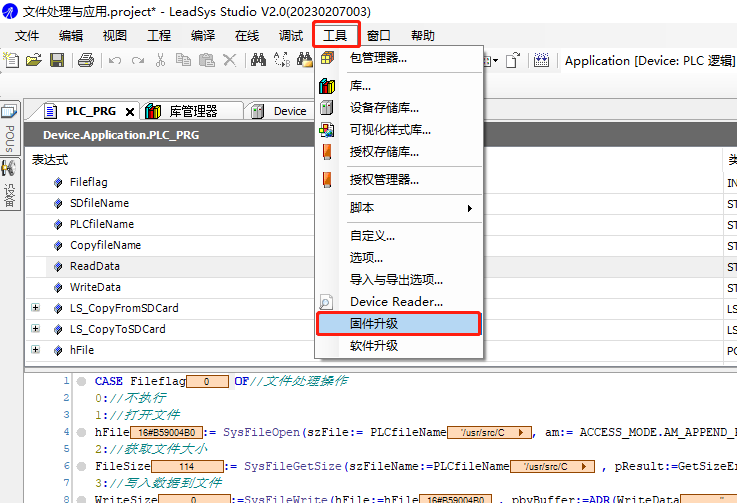
SC2-C系列PLC只能识别FAT32格式最大32G的U盘和SD卡，SD卡可通过带tyep-C接头的读卡器插到SC2-C上。

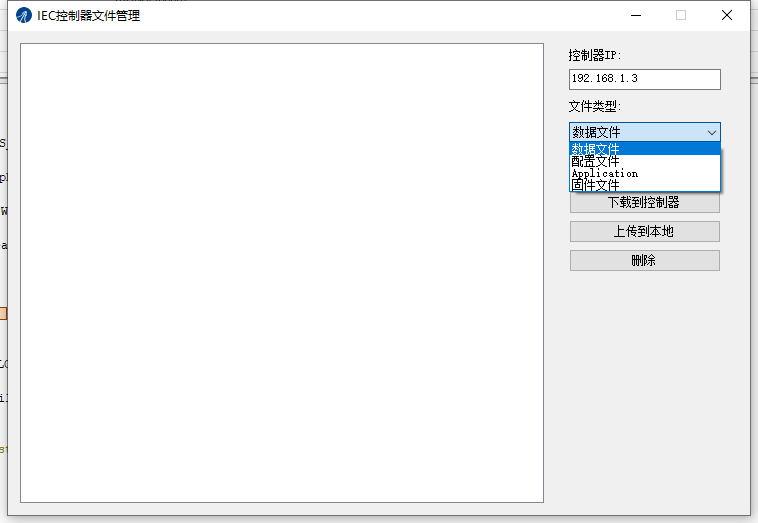
一般情况下建议用户将系统类的参数文件存储在“数据文件”文件夹内，工艺类的文件存储在“配置文件”内。文件路径如下。若操作的文件无路径仅有文件名时，则默认操作“数据文件”目录下的文件。

数据文件路径：/usr/src/CODESYSControl/UsrData/文件名称

配置文件路径：/usr/src/CODESYSControl/UsrConfig/文件名称

SC2-C系列PLC的内部文件可在“IEC控制器文件管理”中查看，打开方式如下图图所示。





使用文件处理功能前须在工程中添加“SysFile”、“SysTypes Interfaces”和“CmpErrors”库文件。

本文介绍以文件的读写操作和文件复制、重命名、删除操作两个例程介绍文件处理。

1. **文件处理指令**

文件处理指令一览，这些指令均为FC，可以在程序中直接调用，不需要再程序中实例化。

|  |  |  |
| --- | --- | --- |
| **指令类别** | **名称** | **功能** |
| 文件处理指令 | SysFileOpen | 打开指定文件 |
| SysFileClose | 关闭指定文件 |
| SysFileCopy | 复制指定文件 |
| SysFileDelete | 删除指定文件 |
| SysFileRename | 重命名指定文件 |
| SysFileWrite | 将数据内容写入文件 |
| SysFileRead | 读取文件的数据内容 |
| SysFileSetPos | 设置文件光标到指定位置 |
| SysFileGetPos | 获取文件光标的位置 |
| SysFileGetSize | 获取文件大小 |
| SysFileGetTime | 获取文件时间 |

* + - 1. **打开指定文件SysFileOpen**

该指令用于打开或创建指定路径下文件，指令返回为文件句柄，可作为读写文件等指令操作的输入。其格式如下。

Return:=SysFileOpen(szFile:=文件名,am:=文件打开模式,pResult:=文件打开错误码指针);

其中：

文件打开模式类型：

AM\_READ：以只读权限打开现有文件。若文件不存在，则打开失败；

AM\_WRITE：创建具有写权限的新文件。若文件存在，则取消创建新文件；

AM\_APPEND：以追加(仅写)权限打开现有文件。若文件不存在，则打开失败；

AM\_READ\_PLUS：以读/写访问权限打开现有文件。若文件不存在，则打开失败；

AM\_WRITE\_PLUS：创建具有读/写权限的新文件。若文件存在，则取消创建新文件；

AM\_APPEND\_PLUS：以追加(读/写)访问权限打开现有文件。若文件不存在，将创建新文件。

文件打开模式类型中\_PLUS模式，在文件读取、写入后都需要调用SysFileGetPos或SysFileSetPos获取光标位置，否则文件光标可能在一个无效的位置，会导致读取、写入失败。

将U盘内文件数据复制到PLC内部的文件时只具有只读权限，若需要更多操作权限建议使用AM\_WRITE\_PLUS模式在PLC内部创建具有读/写权限的新文件，再将U盘内文件数据复制到创建的内部文件中。

* + - 1. **关闭指定文件SysFileClose**

SysFileClose指令用于文件读写操作结束后关闭文件，指令的返回值为关闭文件错误码。文件处理完成后必须要关闭文件，否则会导致内存溢出等问题，严重可能导致PLC系统崩溃。其格式如下。

Return:=SysFileClose(hFile:= 文件句柄);

* + - 1. **复制指定文件SysFileCopy**

SysFileCopy指令可以将szSourceFileName的文件的内容复制到szDestFileName的文件中, 指令的返回值为复制文件错误码。其格式如下。

Return:=SysFileCopy(szDestFileName:=目标文件路径及文件名,szSourceFileName:=源文件路径

及文件名,pulCopied:=操作的字节数);

其中szDestFileName中的文件夹路径必须是已经存在的，否则将会报错。其中若文件存在，则复制之后会覆盖原文件，若文件不存在则复制之后会创建文件。CoSourceFileName中的文件必须是存在的，否则将会报错。

* + - 1. **删除指定文件SysFileDelete**

SysFileDelete指令用于删除指定文件，指令的返回值为错误码。其格式如下。

SysFileDelete(szFileName:= 指定文件路径及文件名);

* + - 1. **将数据内容写入文件SysFileWrite**

SysFileWrite指令用于在文件打开后，写入数据内容，指令的返回值为实际写入文件字节个数。其格式如下。

Return:=SysFileWrite(hFile:= 文件句柄, pbyBuffer:= 写入数据变量的指针, ulSize:= 写入的

字节数, pResult:= 错误码指针);

* + - 1. **读取文件的数据内容SysFileRead**

SysFileRead指令用于在文件打开后，读取文件的数据内容，指令的返回值为实际读取文件字节个数。其格式如下。

Return:=SysFileRead(hFile:= 文件句柄,pbyBuffer:=读取数据变量的指针,ulSize:=读取的字节

数, pResult:= 错误码指针);

* + - 1. **设置文件光标到指定位置SysFileSetPos**

SysFileSetPos指令用于设置文件光标到指定位置，指令的返回值为错误码。其格式如下。

Return:=SysFileSetPos(hFile:= 文件句柄, ulOffset:= 偏移位置);

* + - 1. **获取文件光标的位置SysFileGetPos**

SysFileGetPos指令用于读取文件光标当前位置，指令的返回值为错误码。其格式如下。

Return:=SysFileGetPos(hFile:= 文件句柄 , pulPos:=光标位置 );

* + - 1. **获取文件大小SysFileGetSize**

SysFileGetSize指令用于获取指定文件的大小，指令的返回值为获取指定文件的字节个数。其格式如下。

Return:=SysFileGetSize(szFileName:=文件名, pResult:= 错误代码的指针);

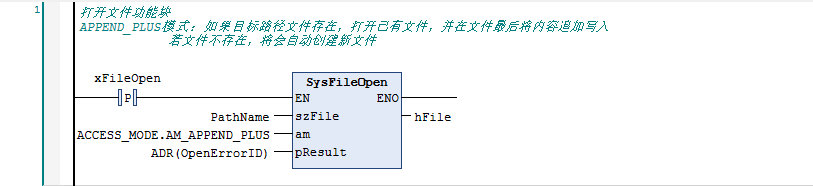
* + - 1. **获取文件时间SysFileGetTime**

SysFileGetTime指令用于获取指定文件的文件时间，指令的返回值为错误码。其格式如下。

Return:=SysFileGetTime(szFileName:= 文件句柄, ptFileTime:= 文件时间变量的指针);

1. **读写文件**
   1. 打开文件

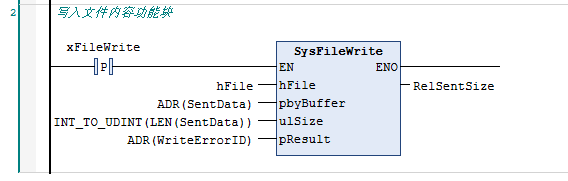
调用SysFileOpen函数。szFile为文件路径和文件名，是字符串类型，这里填入变量PathName初始化值为'/usr/src/CODESYSControl/UsrData/test1.txt'；am为文件的打开模式，这里使用APPEND\_PLUS模式；pResult是错误指针变量。



当xFileOpen触发上升沿，打开文件。如果文件打开失败，则hFile输出句柄（空句柄为16#FFFFFFFF）。

* 1. 写入文件内容

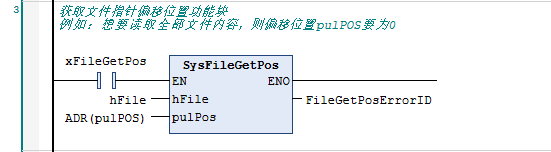
调用SysFileWrite函数。hFile为文件的句柄，由SysFileOpen函数产生；pbyBuffer为写入内容的字符串指针；ulSize为写入内容的字符串长度；pResult是错误指针变量。



当xFileWrite触发上升沿，字符串SentData被加到文件原内容之后。

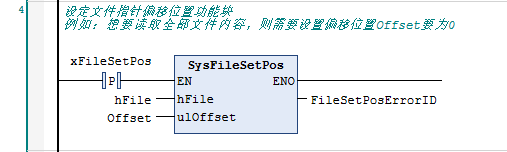
* 1. 读取、设置光标位置

读取光标位置，调用SysFileGetPos函数。hFile为文件的句柄，由SysFileOpen函数产生；pulPos为输出光标位置变量的指针。



当xFileGetPos触发上升沿，获取到光标位置存放在变量pulPOS中。

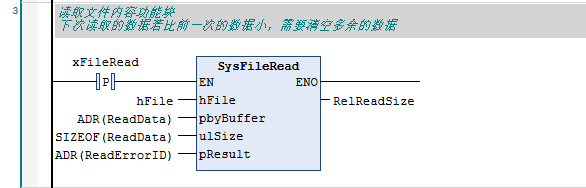
设置光标位置，调用SysFileSetPos函数。hFile为文件的句柄，由SysFileOpen函数产生；ulOffset为设置光标的位置。



当xFileSetPos触发上升沿，光标移动到Offset位置。

* 1. 读取光标后文件内容

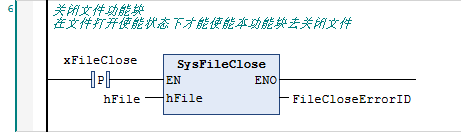
调用SysFileRead函数。hFile为文件的句柄，由SysFileOpen函数产生；pbyBuffer为读取内容输出的字符串指针；ulSize为读取到的内容字符串长度；pResult是错误指针变量。



当xFileRead触发上升沿，读取文件光标位置到结束位置的内容，并存放在ReadData字符串变量中。例如数据“This is the first line.$R”；若光标位于‘T’前面，则读取全部数据；若光标位于‘t’和‘h之间’，则从第11个字符开始读取，读取数据内容为“e first line.$R”。如果读取到的数据为空，可能是光标在文件结束位置所致，应调用SysFileSetPos函数设置光标位置。

* 1. 关闭文件

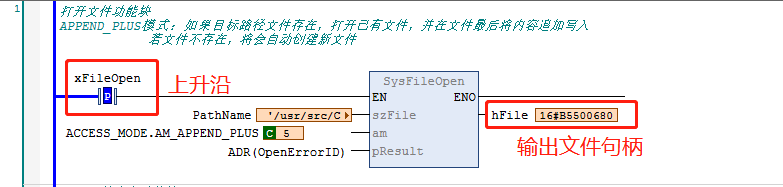
调用SysFileClose函数。hFile为文件的句柄，由SysFileOpen函数产生。



当xFileClose触发上升沿，关闭文件。文件处理完成后必须要关闭文件，否则会导致内存溢出等问题，严重可能导致PLC系统崩溃。如果文件未打开，调用SysFileClose函数会导致程序跑死。

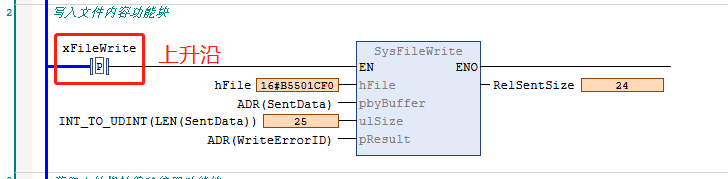
* 1. 写入、读取文件例程使用指南

①打开文件。xFileOpen触发上升沿，打开文件，产生文件句柄。

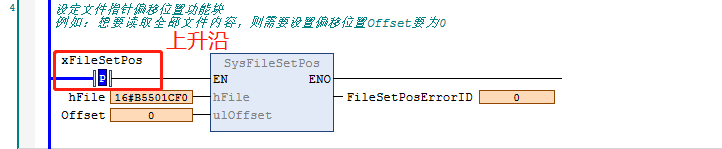


②将SentData字符串写入文件。当xFileWrite触发上升沿，SentData字符串被写入文件。

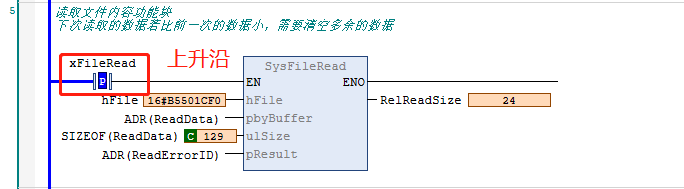




③移动光标到文件开头。xFileSetPos触发上升沿，将移动光标到文件开头。

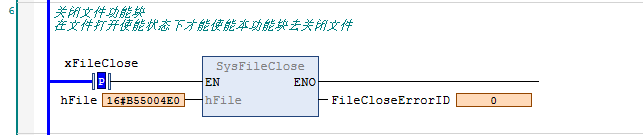


④读取文件内容。xFileRead触发上升，数据读取存放到pulPOS中。



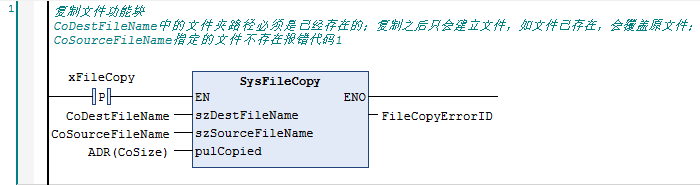


⑤关闭文件。xFileClose触发上升沿，关闭文件。



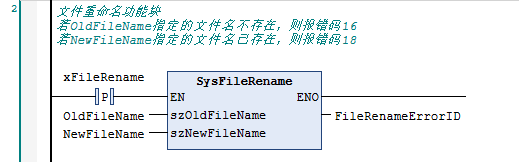
1. **复制、重命名、删除文件**
   1. 复制文件

调用SysFileCopy函数。szDestFileName为复制后的文件路径和文件名，szSourceFileName为被复制的文件路径和文件名，pulCopied为复制文件大小的指针变量。



* 1. 重命名文件

调用SysFileRename函数。szOldFileName为原文件路径和文件名，szNewFileName为重命名后的文件路径和文件名。



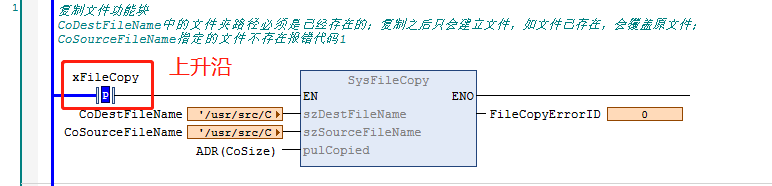
* 1. 删除文件

调用SysFileDelete函数。szFileName为需删除文件的文件路径和文件名。

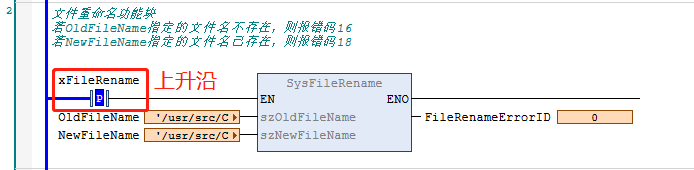


* 1. 复制、重命名、删除文件例程使用指南

①复制文件。xFileCopy触发上升沿，将2.6新建的文件'/usr/src/CODESYSControl/UsrData/test1.txt'，复制为'/usr/src/CODESYSControl/UsrData/test1\_copy.txt'。



②重命名文件。xFileRename触发上升沿，将①复制出来的文件重命名为'/usr/src/CODESYSControl/UsrData/test2.txt'。



③删除文件。xFileDelete触发上升沿，将'/usr/src/CODESYSControl/UsrData/test1.txt'文件删除。

