Application Note

File I/O Operations in Structured Text with ASIC-100/200

ASIC-100/200 now supports reading and writing data structures to data files on the hard disk and performing file operations including copying files over a network using Structured Text operations.

These operations are as follows:

```
NEWFILE( fcb, FILE := fnamestring );
OPENFILE( fcb, FILE := fnamestring );
READFILE( fcb, OUT := structure );
WRITEFILE( fcb, IN := structure );
READFILE( fcb, OUT := structure, F := anystring, S := anystring, T := anystring );
WRITEFILE( fcb, IN := structure, F := anystring, S := anystring, T := anystring );
CLOSEFILE( fcb );
REWINDFILE( fcb );
DELETEFILE( fcb, IN := fnamestring );
COPYFILE( fcb, OUT := tofnamestring, IN := fromfnamestring );
```

Each of the file operations takes as the first parameter the name of a File Control Block (fcb). This File Control Block controls the file operations and provides status as to the progress and outcome of the file operations. When the fcb name is first used in the program, a File Control Block is created with that name. **Make sure that the fcb name that you use will not conflict with other local or global variable names in your program.**

The fcb provides several flags and variables that can be used in the program to determine the outcome of the file operation:

fcb.BUSY	is a BOOL that indicates that the fcb is busy performing an operation,
fcb.OPEN	is a BOOL that indicates that the file has been opened successfully,
fcb.RDN	is a BOOL that indicates that a read operation has been competed,
fcb.WDN	is a BOOL that indicates that a write operation has been completed,
fcb.CLSD	is a BOOL that indicates that the file has been closed,
fcb.EOF	is a BOOL that indicates the End Of File has been seen during reading,
fcb.EFLAG	is a BOOL that indicates a file operation error has occurred
fcb.ERR	is an INT that contains an error code for the operation.

To perform any reading or writing to a data file the file must first be opened, the read and/or write operations performed, and then the file must be closed. The NEWFILE operation will create a new file with the name fnamestring if one does not exist, or if one does exist it will open it and delete the contents. OPENFILE will open an existing file with the name fnamestring, or fail if the file is not found. The fnamestring parameter can be any string variable or constant. If fnamestring does not contain the entire path to the file, the path to the current project directory will be prefixed to it automatically by the fcb. The CLOSEFILE command is used to close an

open file. The CLOSEFILE command does not need a filename because it will get it from the specified fcb.

READFILE and WRITEFILE take the name of a data structure (STRUCT) variable to read into or to write from. Optional forms of the READFILE and WRITEFILE operations can be used to additionally specify strings of characters to be used between structure members (field separator F), to delimit STRING values (string delimiter S), and to be used at the end of the line (end of line terminator T). These format strings are stored in the fcb and can be changed by the READFILE and WRITEFILE commands. The default values for these strings are: F:= " " (space), S := "$"" (double quote), T := "$n" (newline). F can be changed to "," for comma separated format.

The REWINDFILE command can be used to reposition the internal file position pointer to the beginning of the file so that the next READFILE operation will begin from the start of the file. This operation is automatically performed when the file is opened.

A file can be deleted using the DELETEFILE command. The name of the file to be deleted is specified in the fnamestring.

A one file can be copied to another file name or to another place on the network using the COPYFILE command. The name of the file to be copied from is specified in the fromfnamestring, and the name of the file to be copied to is specified in the tofnamestring. If you want to copy a file to another directory or somewhere else on the network, specify the path to the destination file in the tofnamestring.

Each of the File I/O operations may fail for some reason or other. If this happens, the fcb.EFLAG boolean is set and the fcb.ERR integer receives a value indicating the source of the problem before the fcb.BUSY flag is cleared. These values include the following:

FILE_FCB_BUSY	15
NO_FILENAME_SPECIFIED	16
FILE_NOT_OPENED	17
FILE_NOT_FOUND	18
DISK_FULL	19
READ_FAILED	20
FILE_COPY_FAILED	21
WRITE_FAILED	22
EOL_UNEXPECTED	23
EOF_UNEXPECTED	24

For more details on these error values, consult the Program Editor help file for File I/O functions.

When File I/O operations are programmed within a Step in a SFC diagram, the commands following in the Step can not be executed and the step can not be terminated until the File operation has completed. This delay in execution is handled automatically by the ASIC-100

system. For each File I/O command, a worker thread is created to perform the file operation. Only one operation can be performed for each File Control Block at a time. The execution of the SFC Step will be suspended until the worker thread completes and the fcb.BUSY flag is reset. Any other logic in the SFC program outside of the Step containing the File operation will execute normally.

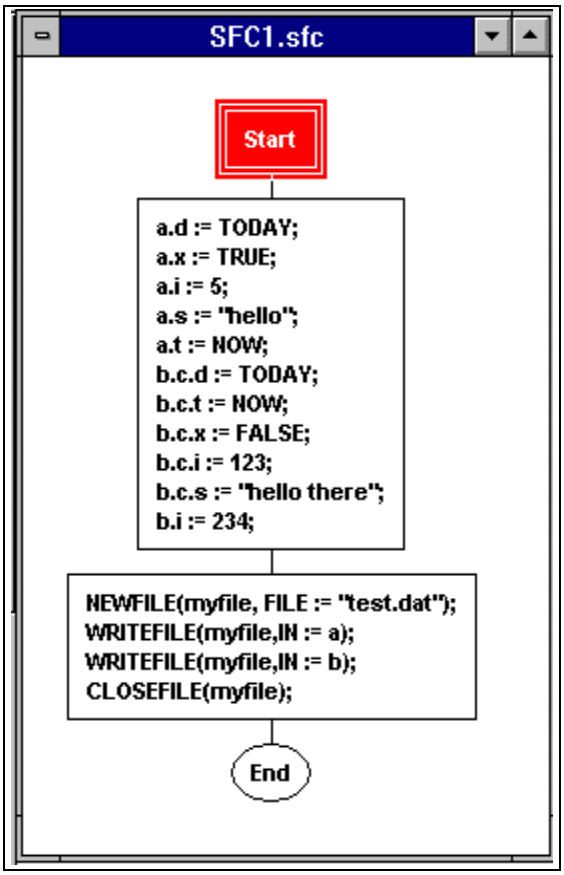
```
TYPE
        struct1 : STRUCT
                d : DATE;
                t : TOD;
                x : BOOL;
                i : INT;
                s : STRING;
        END_STRUCT;
        struct2 : STRUCT
                c : struct1;
                i : INT;
        END_STRUCT;
END_TYPE
VAR_GLOBAL
        a : struct1;
        b : struct2;
END_VAR
```

Sample Program

Assume that you have used the Global Symbol Manager to define two data structure types similar to those shown in the above figure. Assume also that you have defined two data structure variables a and b as shown. If you create a SFC program as shown in the figure below and run it, the resulting contents of file ‘test.dat’ will be as follows:

1995-09-18 16:24:03 TRUE 5 "hello"
1995-09-18 16:24:03 FALSE 123 "hello there" 234

This sample program uses the default file format strings, which you can modify if you wish.



ASAP Inc. 100 North Main Street, Suite 235, Chagrin Falls, OH 44022
Telephone: (440) 247-9216 Fax: (440) 247-9218 Email: support@asapinc.com