

```

1      //This Function block requires SysFile = SysFile, 3.5.6.0 in the Library
      Manager
2      FUNCTION_BLOCK TEXT_TO_FILE
3      VAR_INPUT
4          xFilePath : STRING ;
5          xText_Write : STRING ( 2000 ) ;
6          xAppend_OverWrite : BOOL ;
7          xAppend_Pos : UDINT ;
8          xWrite : BOOL ;
9          xRead : BOOL ;
10         xLine_No_To_Read : UDINT ;
11     END_VAR
12     VAR_OUTPUT
13         xText_Read : STRING ( 2000 ) ;
14         xDone : BOOL ;
15     END_VAR
16     VAR
17
18     // Write Variables
19         xWrite_Trig : R_TRIG ;
20         xStart_Write : BOOL ;
21         xNew_Line : STRING ( 2000 ) ;
22
23     // Read Variables
24         xRead_Trig : R_TRIG ;
25         xStart_Read : BOOL ;
26         xLine_Read : UDINT ;
27
28         hFile          : RTS_IEC_HANDLE := RTS_INVALID_HANDLE ;
29         udiBytesRead    : UDINT ;
30         udiBytesCopied  : UDINT ;
31         udiWrite        : UDINT ;
32         udiFileSize     : UDINT ;
33         udiPos          : UDINT ;
34
35     ((* Error Codes *))
36         udiPosError      : RTS_IEC_RESULT ;
37         udiCopyError     : RTS_IEC_RESULT ;
38         udiDeleteError1  : RTS_IEC_RESULT ;
39         udiDeleteError2  : RTS_IEC_RESULT ;
40         udiWriteError1   : RTS_IEC_RESULT ;
41         udiWriteError2   : RTS_IEC_RESULT ;
42         udiWriteError3   : RTS_IEC_RESULT ;
43         udiOpenError1    : RTS_IEC_RESULT ;
44         udiOpenError2    : RTS_IEC_RESULT ;
45         udiOpenError3    : RTS_IEC_RESULT ;
46         udiOpenError4    : RTS_IEC_RESULT ;
47
48         udiReadError1    : RTS_IEC_RESULT ;
49         udiReadError2    : RTS_IEC_RESULT ;
50         udiReadError3    : RTS_IEC_RESULT ;
51         udiCloseError1   : RTS_IEC_RESULT ;
52         udiCloseError2   : RTS_IEC_RESULT ;
53         udiCloseError3   : RTS_IEC_RESULT ;
54         udiCloseError4   : RTS_IEC_RESULT ;
55         udiSizeError     : RTS_IEC_RESULT ;
56         udiDirDeleteError : UDINT ;

```

```
57         udiCreateError : UDINT ;
58     END_VAR
59
1   xWrite_Trig ( ) ;
2   xWrite_Trig . CLK := xWrite ;
3   xStart_Write := xWrite_Trig . Q ;
4
5   xRead_Trig ( ) ;
6   xRead_Trig . CLK := xRead ;
7   xStart_Read := xRead_Trig . Q ;
8
9   IF xStart_Write THEN
10      xDone := FALSE ;
11      IF xAppend_OverWrite THEN
12         hFile := SysFileOpen ( szFile := xFilePath , am := ACCESS_MODE . AM_APPEND
, pResult := ADR ( udiOpenError1 ) ) ;
13      END_IF
14      IF xAppend_OverWrite = FALSE THEN
15         hFile := SysFileOpen ( szFile := xFilePath , am := ACCESS_MODE .
AM_WRITE_PLUS , pResult := ADR ( udiOpenError1 ) ) ;
16      END_IF
17      udiPosError := SysFileGetPos ( hFile := hFile , pulPos := ADR ( udiPos ) ) ;
18      IF hFile <> RTS_INVALID_HANDLE THEN
19         xNew_Line := concat ( '$n' , xText_Write ) ;
20         IF udiPos = 16#0 THEN
21            udiWrite := SysFileWrite ( hFile := hFile , pbyBuffer := ADR (
xText_Write ) , ulSize := INT_TO_UDINT ( LEN ( xText_Write ) ) , pResult := ADR
( udiWriteError1 ) ) ;
22         ELSE
23            udiWrite := SysFileWrite ( hFile := hFile , pbyBuffer := ADR (
xNew_Line ) , ulSize := INT_TO_UDINT ( LEN ( xNew_Line ) ) , pResult := ADR (
udiWriteError1 ) ) ;
24         END_IF
25         udiCloseError1 := SysFileClose ( hFile := hFile ) ;
26      END_IF
27      xDone := TRUE ;
28   END_IF
29
30   IF xStart_Read THEN
31      xDone := FALSE ;
32      hFile := SysFileOpen ( szFile := xFilePath , am := ACCESS_MODE . AM_READ ,
pResult := ADR ( udiOpenError1 ) ) ;
33      IF hFile <> RTS_INVALID_HANDLE THEN
34         udiBytesRead := SysFileRead ( hFile := hFile , pbyBuffer := ADR (
xText_Read ) , ulSize := SIZEOF ( xText_Read ) , pResult := ADR ( udiReadError1 )
) ;
35         xText_Read [ udiBytesRead ] := 0 ;
36      END_IF
37   END_IF
38
39
```