Logger TLM

Table of contents

1 About

1.1 Overview

1.1.1 Introduction

The **LOGGER** TLM is a free firmware add-on option for SoftPLC version 4.x and later that adds the capability of logging a group of data values to comma-separated values (CSV) text files. This capability is implemented as a TOPDOC Loadable Module (TLM), written in C++ and implements a ladder instruction (TLI) which triggers the log interval. This document describes the installation, usage, and functionality of the LOGGER TLM.

1.1.2 Concepts

The **SoftPLC runtime engine** software supports TLM's, which are shared library extensions to SoftPLC. A TLM may be loaded either as a **DRIVER** or as a **MODULE**. The difference between a DRIVER and a MODULE is that a DRIVER is called once per SoftPLC scan, and optionally an additional number of times per scan. A MODULE is only called when the control program decides to call it and not as an inherent part of the scan. TLM's are made known to SoftPLC in the MODULES.LST file which may be edited by **TOPDOC NexGen** by traversing to: PLC | Modules.

This LOGGER TLM is to be used as a MODULE, not as a DRIVER. It provides three (3) TOPDOC Loadable Instructions (TLI's) to trigger the log interval and copy the log files to other partitions or media.

1.1.3 Features

This TLM takes a configuration file named **LOGGER.LST** which holds the disk base directory for all log (CSV) files and one or more data groups. A data group contains a list of the word or bit level datatable addresses or tagnames to log and an optional time format specifier. Each data group must have a unique name.

All log files have a fixed time base of one week and will be named with the following convention where “nn” is the week number.

```
[diskbase]/week-nn-[group].csv
```

Week numbers will be generated in the range 1 to 53 per ISO 8601:1988. See Appendix A.

1.1.4 Requirements

- SoftPLC version 4.x runtime
- Dedicated Compact Flash partition for “diskbase” (ie. /dev/sda3 mounted as /storage)

1.2 Warranty

1.2.1 Terms of Use

Because of the variety of uses of the information described in this manual, the users of, and those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the information. In no event will SoftPLC Corporation be responsible or liable for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

SOFTPLC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

SoftPLC Corporation reserves the right to change product specifications at any time without notice. No part of this document may be reproduced by any means, nor translated, nor transmitted to any magnetic medium without the written consent of SoftPLC Corporation.

SoftPLC, and TOPDOC are registered trademarks of SoftPLC Corporation.

First Printing: February, 2011
Latest Printing: February, 2011

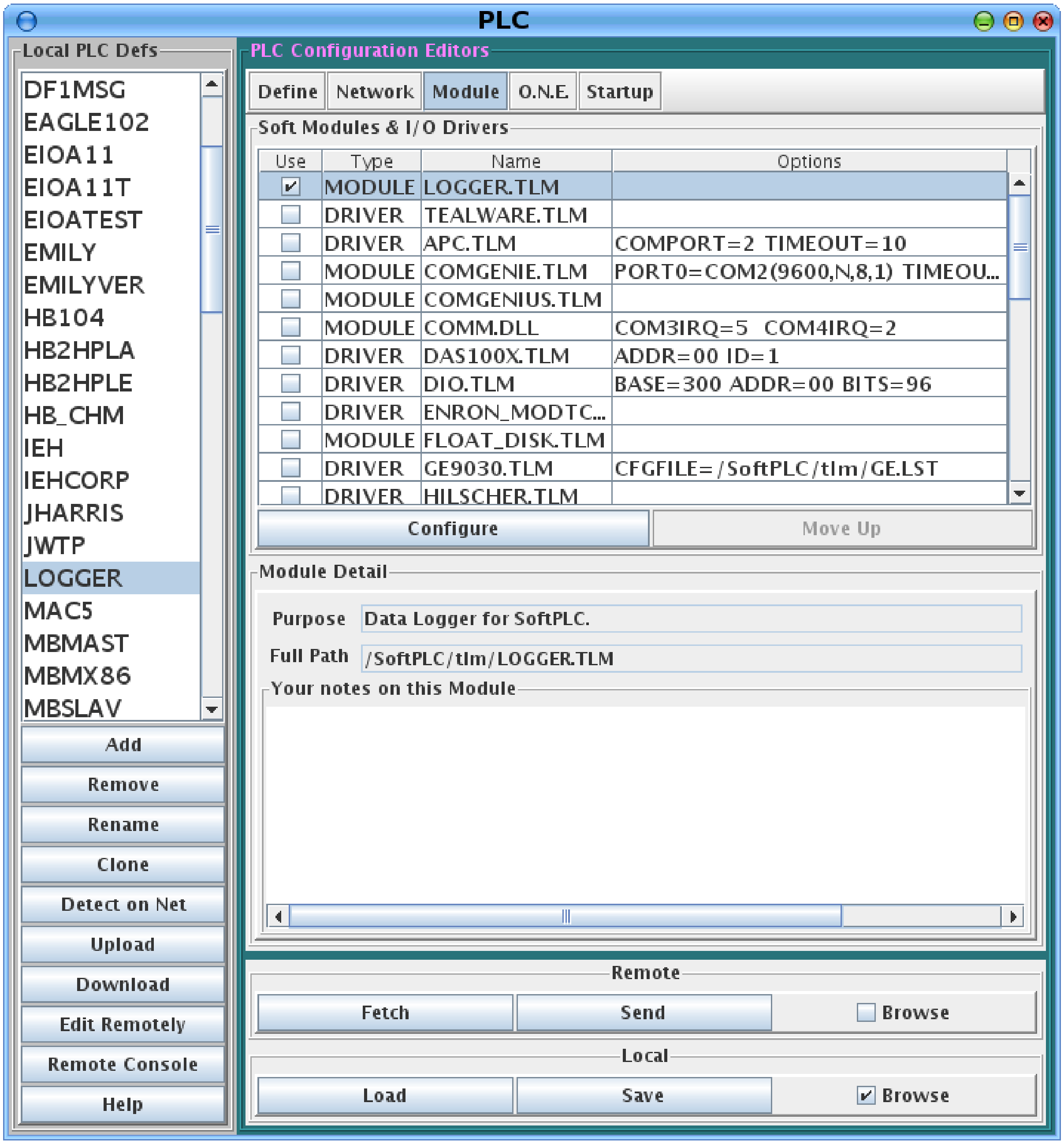
SoftPLC Corporation
25603 Red Brangus Drive
Spicewood, Texas 78669
USA Telephone: 1-800-SoftPLC
Fax: 512/264-8399
URL: http://softplc.com
Email: support@softplc.com

2 Installation and Usage

2.1 Installation

2.1.1 Installation

The TLM is named **logger.tlm.so**, and the configuration file is **LOGGER.LST**. You will need to copy these onto the SoftPLC flash in the /SoftPLC/tlm directory and the TLM to your PC's hard drive in the \SoftPLC\tlm folder. Then, to use it you have to enable and configure it using **TOPDOC NexGen's PLC | MODULES** editor. Select *Use* for LOGGER.TLM, then click on the Configure button to edit the LOGGER.LST file.



DF1MSG
EAGLE102
EIOA11
EIOA11T
EIOATEST
EMILY
EMILYVER
HB104
HB2HPLA
HB2HPLE
HB_CHM
IEH
IEHCORP
JHARRIS
JWTP
LOGGER
MAC5
MBMAST
MBMX86
MBSLAV

Use	Type	Name	Options
☑	MODULE	LOGGER.TLM	
□	DRIVER	TEALWARE.TLM	
□	DRIVER	APC.TLM	COMPORT=2 TIMEOUT=10
□	MODULE	COMGENIE.TLM	PORT0=COM2(9600,N,8,1) TIMEOU...
□	MODULE	COMGENIUS.TLM	
□	MODULE	COMM.DLL	COM3IRQ=5 COM4IRQ=2
□	DRIVER	DAS100X.TLM	ADDR=00 ID=1
□	DRIVER	DIO.TLM	BASE=300 ADDR=00 BITS=96
□	DRIVER	ENRON_MODTC...	
□	MODULE	FLOAT_DISK.TLM	
□	DRIVER	GE9030.TLM	CFGFILE=/SoftPLC/tlm/GE.LST
□	DRIVER	HILSCHER.TLM	

Purpose: Data Logger for SoftPLC.

Full Path: /SoftPLC/tlm/LOGGER.TLM

2.2 Configuration

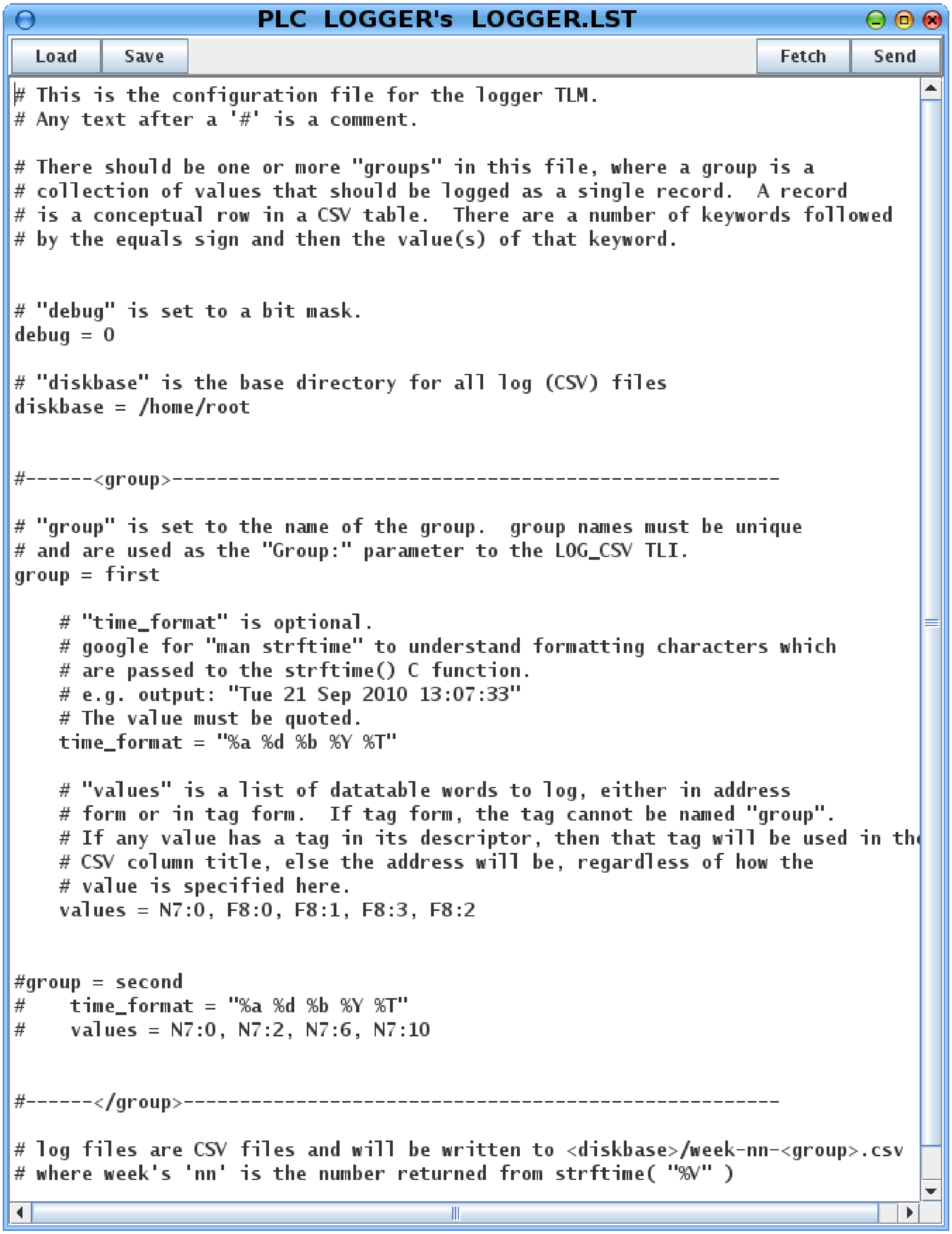
2.2.1 LOGGER.LST Configuration file

The configuration file for the LOGGER TLM is **/SoftPLC/tlm/LOGGER.LST**. A sample is shown below, which illustrates the use of a variety of datatable addresses and tagnames. The comments in the file explain the file format. See Appendix B for an example of the CSV log file that will be created based on this configuration.

```
# This is the configuration file for the logger TLM.
# Any text after a '#' is a comment.

# There should be one or more "groups" in this file, where a group is a
# collection of values that should be logged as a single record.  A record
# is a conceptual row in a CSV table.  There are a number of keywords followed
# by the equals sign and then the value(s) of that keyword.

# "debug" is set to a bit mask.
debug = 0

# "diskbase" is the base directory for all log (CSV) files
diskbase = /storage

#------(group)------------------------------------------------------
# "group" is set to the name of the group.  group names must be unique
# and are used as the "Group:" parameter to the LOG_CSV TLI.
group = first

    # "time_format" is optional.
    # google for "man strftime" to understand formatting characters which
    # are passed to the strftime() C function.
    # e.g. output: "Tue 21 Sep 2010 13:07:33"
    # The value must be quoted.
    time_format = "%a %d %b %Y %T"

    # "values" is a list of datatable words to log, either in address
    # form or in tag form.  If tag form, the tag cannot be named "group".
    # If any value has a tag in its descriptor, then that tag will be used in the
    # CSV column title, else the address will be, regardless of how the
    # value is specified here.
    values = N7:0, F8:0, F:8:3, I:0/0, R6:0.POS, LASTLOGERROR

#group = second
#    time_format = "%a %d %b %Y %T"
#    values = N7:0, N7:2, N7:6, N7:10
#------(/group)-----------------------------------------------------

# log files are CSV files and will be written to [diskbase]/week-nn-[group].csv
# where week's 'nn' is the number returned from strftime( "%V" )
```

2.3 Usage

2.3.1 Editor Usage



```
# This is the configuration file for the logger TLM.
# Any text after a '#' is a comment.

# There should be one or more "groups" in this file, where a group is a
# collection of values that should be logged as a single record.  A record
# is a conceptual row in a CSV table.  There are a number of keywords followed
# by the equals sign and then the value(s) of that keyword.


# "debug" is set to a bit mask.
debug = 0

# "diskbase" is the base directory for all log (CSV) files
diskbase = /home/root


#------<group>----------------------------------------------------------

# "group" is set to the name of the group.  group names must be unique
# and are used as the "Group:" parameter to the LOG_CSV TLI.
group = first

    # "time_format" is optional.
    # google for "man strftime" to understand formatting characters which
    # are passed to the strftime() C function.
    # e.g. output: "Tue 21 Sep 2010 13:07:33"
    # The value must be quoted.
    time_format = "%a %d %b %Y %T"

    # "values" is a list of datatable words to log, either in address
    # form or in tag form.  If tag form, the tag cannot be named "group".
    # If any value has a tag in its descriptor, then that tag will be used in the
    # CSV column title, else the address will be, regardless of how the
    # value is specified here.
    values = N7:0, F8:0, F8:1, F8:3, F8:2


#group = second
#    time_format = "%a %d %b %Y %T"
#    values = N7:0, N7:2, N7:6, N7:10


#------</group>---------------------------------------------------------

# log files are CSV files and will be written to <diskbase>/week-nn-<group>.csv
# where week's 'nn' is the number returned from strftime( "%V" )
```

Load button will load the configuration file from the development system's disk.

Save button will write the configuration file to the development system's disk.

Fetch button will load the configuration file from the runtime system's disk.

Send button will write the configuration file to the runtime system's disk. The next step is to restart or cycle power on the SoftPLC for the changes to take place. As an alternative to cycling power, you may enter "Remote Program" mode using NexGen, then select "Remote Program" a second time. This psuedo transition from Remote Program to Remote Program is a signal to the TLM that it should reload its configuration file. This way you can reconfigure without cycling power, although it does require you enter "Remote Program" mode (twice!).

2.3.2 Ladder Instructions

2.3.2.1 1) LOG_CSV

This instruction is used to write a row in a CSV file. This instruction is edge triggered and therefore will log once when the rung goes from false to true. There are three (3) parameters.

“Control:” - Control element (R) address. If log is successful, the Control .CTL Enable Bit (EN, bit 15) will be set true and the Control “.LEN” word will be set to zero. If an error occurs, the Control “.CTL” word error bit (ER, bit 11) will be set true, the Control “.POS” word value will be incremented, and the Control “.LEN” word will contain the integer value of the error. The Control “.CTL” and “.LEN” words are cleared when the rung is no longer true but the “.POS” value is retained. The possible values and respective meanings are:

```
0 = Logged OK
2 = Can't write to log file (unable to open)
7 = Can't create log file
8 = Group does not exists in LOGGER.LST
9 = Can't write to log file (disk may be full)
```

“Result:” - String element (ST) that will returns a text string indicating success or failure. The string will correspond directly to the integer value set in the Control “.LEN” word.
“Group:” - String element (ST) that contains the group name to log.

2.3.2.2 2) LOG_FILENAME

This instruction is used to retrieve the current log filename. This instruction is level triggered and therefore will execute when the rung state is true. There are two (2) parameters.

“Group:” - String element (ST) that contains the group name.
“Result:” - String element (ST) that will return the name of the current log file for specified group. name provided.

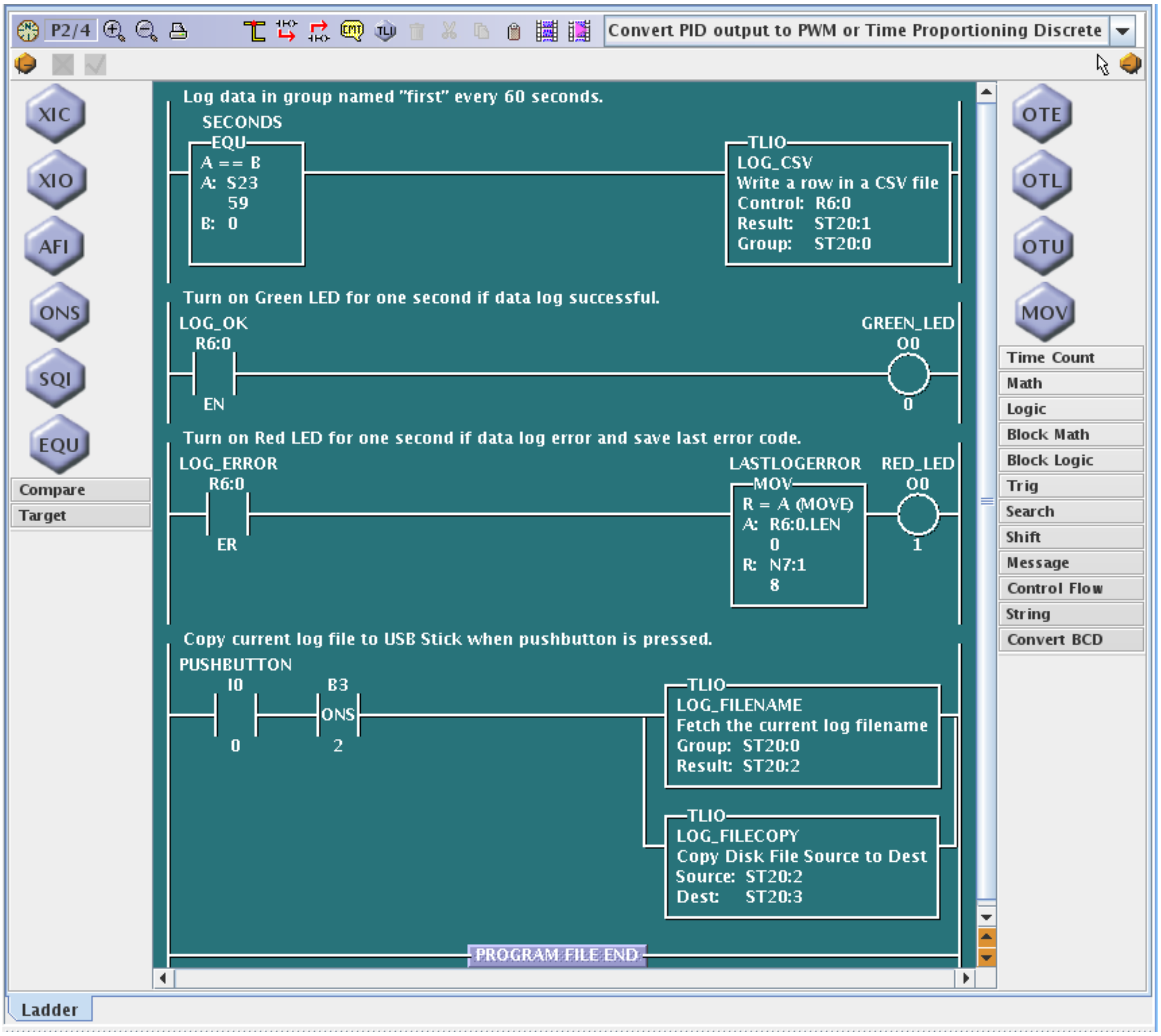
2.3.2.3 3) LOG_FILECOPY

This instruction is used to copy a disk file to another name, such as to a removable media like a USB memory stick. This instruction is level triggered and therefore will execute when the rung state is true. There are two (2) parameters.

“Source:” - String element (ST) that contains the path and filename to be copied.
“Dest:” - String element (ST) that contains the destination path and filename of the copy.

2.3.3 Ladder Example

Log data once every 60 seconds and copy the log file to /usbstick/currentlogfile.csv when a pushbutton connected to I:0/0 is pressed.



Log data in group named "first" every 60 seconds.

```
SECONDS
-EQU-                      -TLIO-
A == B                     LOG_CSV
A: S23                     Write a row in a CSV file
   59                      Control: R6:0
B: 0                       Result:  ST20:1
                           Group:   ST20:0
```

Turn on Green LED for one second if data log successful.

```
LOG_OK                                GREEN_LED
R6:0                                  O0
EN                                    0
```

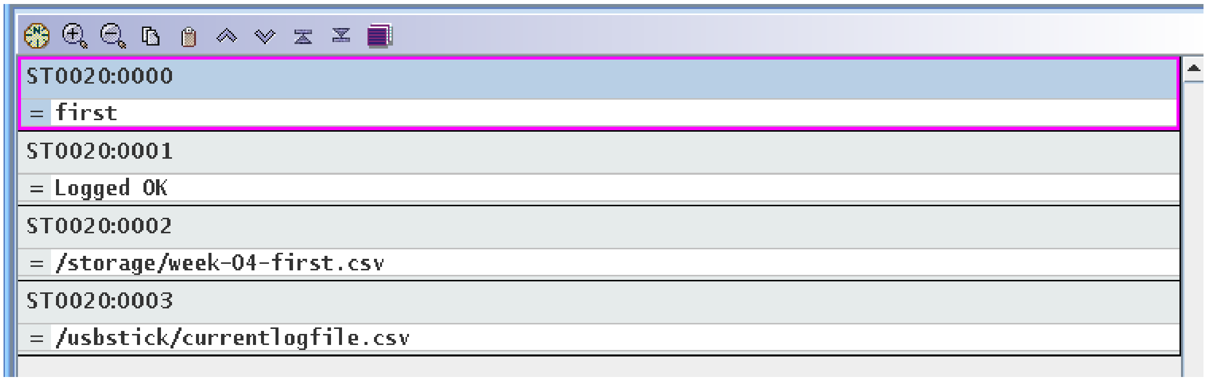
Turn on Red LED for one second if data log error and save last error code.

```
LOG_ERROR                 LASTLOGERROR   RED_LED
R6:0                      -MOV-          O0
ER                        R = A (MOVE)   1
                          A: R6:0.LEN
                             0
                          R: N7:1
                             8
```

Copy current log file to USB Stick when pushbutton is pressed.

```
PUSHBUTTON
I0      B3                 -TLIO-
        ONS                LOG_FILENAME
0       2                  Fetch the current log filename
                           Group:  ST20:0
                           Result: ST20:2

                           -TLIO-
                           LOG_FILECOPY
                           Copy Disk File Source to Dest
                           Source: ST20:2
                           Dest:   ST20:3
```

PROGRAM FILE END



ST0020:0000
= first
ST0020:0001
= Logged OK
ST0020:0002
= /storage/week-04-first.csv
ST0020:0003
= /usbstick/currentlogfile.csv

2.4 Appendix

2.4.1 Appendix A

The LOGGER TLM uses the C function "strftime()" with the %V conversion option to set the week number (nn) in the log filename convention used.

```
[diskbase]/week-nn-[group].csv
```

Function: "strftime()" - format date and time
From the man page:
%V
The ISO 8601:1988 week number of the current year as a decimal number, range 01 to 53, where week 1 is the first week that has at least 4 days in the current year, and with Monday as the first day of the week. See also %U and %W. (SU)
Reference

2.4.2 Appendix B

Example CSV log file generated using LOGGER.LST Configuration file

```
root@SMART[/storage]# cat week-04-first.csv
Time,N0007:0000,F0008:0000,F0008:0003,PUSHBUTTON,R0006:0000.POS,LASTLOGERROR
"Fri 28 Jan 2011 11:24:00",100,1234.432,1.234568e+09,1,1,8
"Fri 28 Jan 2011 11:25:00",100,1234.432,1.234568e+09,1,1,8
"Fri 28 Jan 2011 11:26:00",100,1234.432,1.234568e+09,0,1,8
"Fri 28 Jan 2011 11:27:00",99,1234.432,1.234568e+09,0,1,8
"Fri 28 Jan 2011 11:28:00",99,1234.432,1.234568e+09,0,1,8
"Fri 28 Jan 2011 11:29:00",99,1234.432,1.234568e+09,0,1,8
"Fri 28 Jan 2011 11:30:00",98,1234.432,1.234568e+09,0,1,8
"Fri 28 Jan 2011 11:31:00",97,1234.432,1.234568e+09,0,1,8
"Fri 28 Jan 2011 11:32:00",96,1234.432,1.234568e+09,0,1,8
"Fri 28 Jan 2011 11:33:00",95,1234.432,1.234568e+09,0,1,8
```

3 All