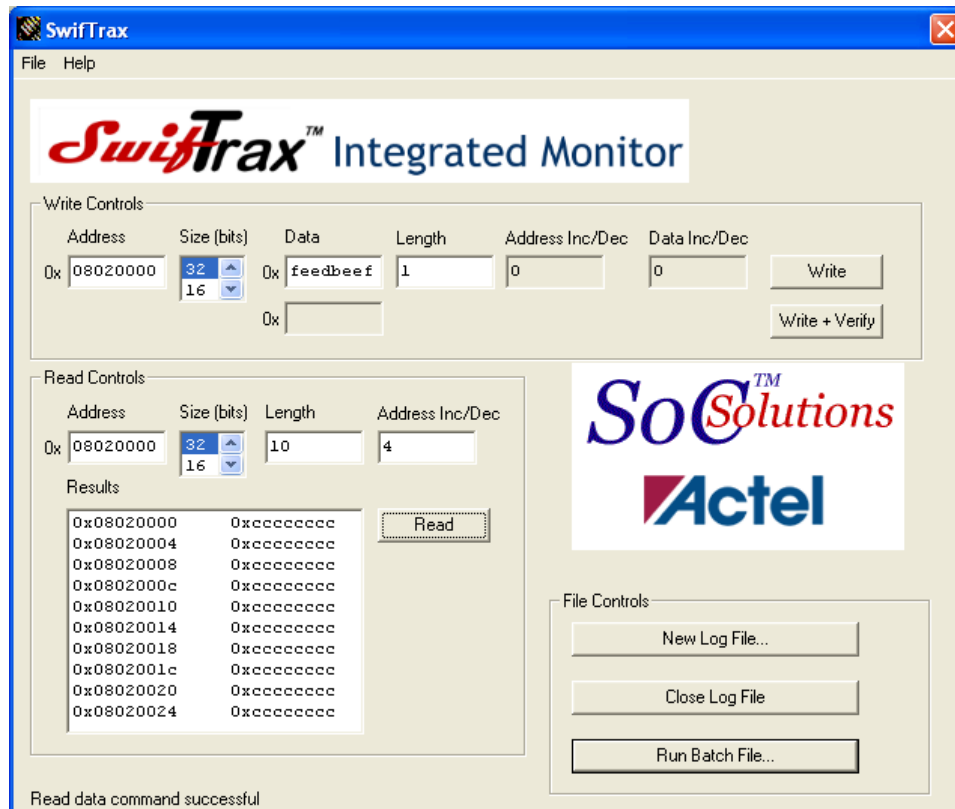


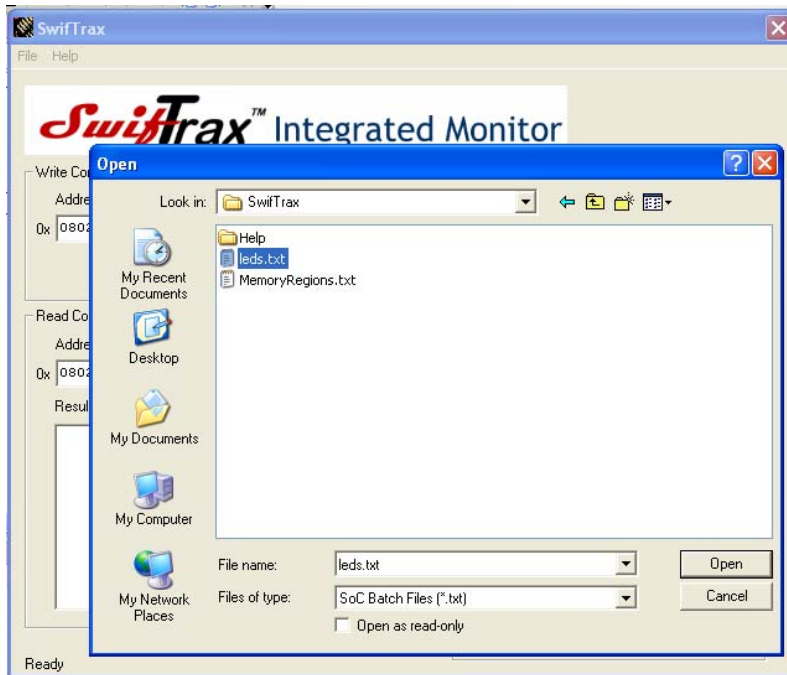
This document describes the commands that can be used in the SwiftTrax Batch mode.



In order to run a batch file, first create the batch file using any text editor. The batch file extension can be ".txt".

To run the batch file, click the "Run Batch File" button on the SwiftTrax Integrated Monitor GUI.

This brings up a standard windows file dialog box.



Select the <batchfile> from and click Open. This will execute the file.

Batch file commands/keywords:

- activatelog
- deactivatelog
- read
- write
- readverify
- writeverify
- readblock
- writeblock
- \$ (comment line)

General Information: Each command must be on a separate line. The keyword always comes first in a command line, and may be followed by some mandatory parameters. Note that any amount of white space may occur at the beginning of the command line, but that some white space (spaces and/or tabs) is mandatory between the successive parameters of a command. Further, a keyword, its parameters, and all white space on a line must not exceed 300 characters. Lastly, a keyword and its parameters must be followed by a newline.

activatelog	"pathname"
Description:	activatelog takes as a parameter a path + filename enclosed in double quotes. activatelog attempts to open the specified file for logging. See also the documentaion for activateLogging.
Example:	activatelog "C:\SoC\testlog.log"

deactivatelog	
Description:	deactivatelog takes no parameters. deactivatelog attempts to close the active logfile.
Example:	Deactivatelog

read	address size
Description:	read takes as parameters an address (hexidecimal) and a size (decimal). These parameters correspond to the first two parameters of readData. read is the batch file equivalent to readData.
Example:	read 01000000 4

write	address size data
Description:	write takes as parameters an address (hexidecimal), a size (decimal), and data (hexidecimal). These parameters correspond to the parameters of writeData. write is the batch file equivalent to writeData.
Example:	write 01000000 4 fc9

readverify	address size verify
Description:	readverify takes as parameters an address (hexidecimal), a size (decimal), and a data value to be verified (hexidecimal). These parameters correspond to the first, second, and fourth parameters of readVerify. readverify is the batch file equivalent to readVerify.
Example:	readverify 1000000 4 fc9

writeverify	address size verify
Description:	writeverify takes as parameters an address (hexidecimal), a size (decimal), and a data value to be verified (hexidecimal). These parameters correspond to the parameters of writeVerify. writeverify is the batch file equivalent to writeVerify.
Example:	writeverify 01000000 4 00000fc8

readblock	address size numberReads addressInc[-]
Description:	readblock reads numberReads memory locations, starting with address, and going through address +- numberReads * addressInc. readblock takes as parameters a base address (hexidecimal), a size (decimal), a number of reads to execute (decimal), and an address increment (decimal). An optional '-' character may be placed directly after addressInc (no white space) if the addressInc should be used as a <i>decrement</i> .
Example:	readblock 01000000 4 256 4 readblock 01000400 4 256 4-

writeblock	address size startData numberWrites addressInc[-] dataInc[-]
Description:	writeblock writes numberWrites memory locations, starting with address, and going through address +- numberWrites * addressInc. Additionally, startData is incremented [decremented] by dataInc. writeblock takes as parameters a base address (hexidecimal), a size (decimal), a startData value (hexadecimal), a number of writes to execute (decimal), an address increment (decimal), and a data increment (decimal). An optional '-' character may be placed directly after addressInc and/or dataInc (no white space) if either should be used as <i>decrements</i> .
Example:	writeblock 01000000 4 ff 256 4 1- writeblock 01000400 4 0 256 4- 1

\$ (comment line)	
Description:	Indicates the line is a comment line. The line will not be processed for commands.
Example:	\$This line is a comment.

Sample batchfile .

```
$write to GPIO0 - connected to LED bank on board.

$set to outputs
write      0c060008      4      000003ff

$write counting pattern

$GPIO clear (turn off all LEDs)
write      0c060004      4      000003ff
$GPIO set (write incrementing pattern)
write      0c060000      4      00000001

write      0c060004      4      000003ff
write      0c060000      4      00000002
write      0c060004      4      000003ff
write      0c060000      4      00000003
write      0c060004      4      000003ff
write      0c060000      4      00000004
```

Logfile Information:

General information: Because the logfile grows very quickly – reads and writes happen thousands of times each second – the disk on which the logfile is located must have at least 4MB of free space available for the application. No new logfiles will be opened on disks having fewer than 4MB free space, and no new entries to the logfile will be made after this limit is reached. Consequently, opening or writing to a logfile on a floppy drive is not allowed – this is by design.

Logfile Format:

(month/day/year)[hour:minute.second.millisecond] {LDWRVYB}{SF} {Data...}

L	=	activateLogging
D	=	deactivateLogging
W	=	writeData
R	=	readData
Y	=	writeVerify
V	=	readVerify
B	=	runBatch

S	=	Success
F	=	Failure

- For L,D,W,R: One entry to the logfile is made.
- For V: Two entries to the logfile are made. The first entry is an R, and tells whether or not the read was successful. The second entry is a V, and tells whether or not the verification was successful.
- For Y: Three entries to the logfile are made. The first entry is a W, and tells whether or not the write was successful. The second is an R, and tells whether or not the read was successful. The third is a Y, and tells whether or not the verification was successful.
- For B: Two entries to the logfile are made. The first entry occurs after the batch file is opened, but before commands are executed. The first entry tells whether or not the batch file was opened successfully. The second entry occurs after all the commands have executed.
The batch file may terminate prematurely if a format error is encountered – the second entry tells whether or not the batch file finished without errors.
- The Data associated with a command:
- For L,D: No data.
- For W,R,V,Y: The first number is the address (hex). The second number is the size (1,2,4 bytes). The third number is the value written(W), read(R), or verified(V,Y).
- For B: The first entry is the batch file name and path. The second entry is the line number of the batch file that caused termination of the batch file.