

Data Logging on the Blue Fusion Model 5222

*How to log data to the controller and export
the resulting file to an ftp server
September, 2005*

5



Control Technology Corporation, Hopkinton, MA • 800.282.5008 • www.ctc-control.com

This application note takes you through an example that logs data to the controller and then exports the file to an ftp server. See http://www.ctc-control.com/customer/techinfo/docs/5200_951/951-520015.pdf for more info on how to set up an ftp server on your network.

It is assumed you have a basic knowledge with script commands used in the BlueFusion. Reference the 5200 Script Language Guide and the Advanced Scripting section of the Model 5200 Logging and FTP Client Application Guide for more information on this. http://www.ctc-control.com/customer/techinfo/docs/5200_951/951-520003.pdf
http://www.ctc-control.com/customer/techinfo/docs/5200_951/951-520015.pdf

Data Logging

- Allows you to periodically record information from the controller in a pre-determined file format.
- Data may consist of any combination of register or data table content, time/date information, and/or constant string text.
- The format of this record information is determined by the definition file called log.ini.
 - This definition file lists all the logging record formats required and each is individually selectable at run-time.
- The data to be logged is saved in NVRAM rather than FLASH since data logging causes a many writes to a file.
 - The mount command is used to create a NVRAM disk. The ramdisk should be virtually linked to the FLASH disk. This allows you to reference the main FLASH disk file structure while actually storing the data in the ram disk.

Here is how to Prepare a NVRAM Disk to store logging files:

1. Open a telnet session: telnet <IP Address>

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Dawn>telnet 12.40.53.47_
```

2. Login to the BlueFusion – User: admin; Password: BlueFusion

```
C:\ Telnet 12.40.53.47
<>** Welcome to the Blue Fusion Administrator Interface **>
Login: admin
Password: *****
BlueFusion/>
```

3. Determine the NVRAM that is available

```
BlueFusion/>memory ram
NURAM 32 Bit 0x04064000-0x041fffff 1687552 Bytes
NURAM 32 Bit 0x04400000-0x045fffff 2097152 Bytes
NURAM 32 Bit 0x04600000-0x047fffff 2097152 Bytes
NURAM 32 Bit 0x04800000-0x049fffff 2097152 Bytes
NURAM 32 Bit 0x04a00000-0x04bfffff 2097152 Bytes
SDRAM 32 Bit 0x062d0000-0x06ffffff 13828096 Bytes
*
BlueFusion/>
```

4. Mount the NVRAM and format for use. The 'mount 0x04400000 4000000 /ramdisk' command mounts a drive called ramdisk of 4000000 Bytes, starting at memory location 0x04400000

```

BlueFusion/>dir
drw-rw-rw-  0 owner group 000256 NOV 29 21:03 _system
Volume: Capacity - 3931920 Free - 3197040 Deleted - 0072960.

BlueFusion/>mount 0x04400000 4000000 /ramdisk
SUCCESS: Mount completed, formatting required if new drive.

BlueFusion/>dir
drw-rw-rw-  0 owner group 000256 NOV 29 21:03 _system
drw-rw-rw-  1 owner group 3749760 DEC 07 15:38 ramdisk
Volume: Capacity - 3931920 Free - 3197040 Deleted - 0072960.

BlueFusion/>cd ramdisk
SUCCESS: cd command successful.

BlueFusion/ramdisk/>dir
Volume: Capacity - 3749760 Free - 3749520 Deleted - 0000000.

BlueFusion/ramdisk/>format disk
Formatting disk...
SUCCESS: Disk formatted successfully.

BlueFusion/ramdisk/>20096=1
20096 = 1

BlueFusion/ramdisk/>reset_

```

5. Mount the virtual directory to store log files

```

BlueFusion/>mount "/_system/Messages/data" "/ramdisk"
SUCCESS: mount completed.

BlueFusion/>dir
drw-rw-rw-  0 owner group 000256 NOV 29 21:03 _system
drw-rw-rw-  1 owner group 3749760 DEC 09 19:08 ramdisk
Volume: Capacity - 3931920 Free - 3197040 Deleted - 0072960.

BlueFusion/>cd /_system/Messages
SUCCESS: cd command successful.

BlueFusion/_system/Messages/>dir
drw-rw-rw-  0 lnked group 000000 JAN 01 00:00 data
Volume: Capacity - 3931920 Free - 3197040 Deleted - 0072960.

BlueFusion/_system/Messages/>

```

Data Logging - Format

The format file, 'log.ini' must reside within the /_system/Messages directory.

This will contain individual records that define the contents to be written to a log file during a logging sequence.

References to registers, data table cell, etc. are defined using an extended format:

Register - %0#dr<register> where # is fixed size for leading zeros and the dr stands for data register

example: %07dr13002 would display register 13002 and fix the format with leading zeros to at least 7 places

Here is a sample log.ini file that creates a header and then logs the date, time, a timer(ms) value, and 3 different register values:

```
log.ini
Date          Time  Timer(ms)  Volts  Lamp1      Lamp2\r\n
%T!MMDDYY!  %T!HH:mm:ss!\t %05dr13002\t %05dr8501\t %01dr1001\t %01dr100
2\r\n
```

The log.ini is simply a text file and can be created with any text editor of your choice.

Data Logging Registers

The logging of data is controlled by two registers:

Log Selection Register (12325) – determines the name of the log file to be written ('Log###.log') where ### is the value stored in register 12325.

Log String Transfer Register (12326) – writing to this register causes the record number from the log.ini file that is written to it to format and write the info to the NVRAM disk file.

The result of a log Operation can be found in:

Log String Result Register (12327) – returns the status of the write. Result values are as follows:

- 0 – Success
- 43 – File error, either 'log.ini' missing or 'Log###.ini' create error.
- 44 – No such record in the 'log.ini' file
- 53 – Write error, check available disk space
- -1 – Default value after setting the Log Selection Register

Snapshot Registers

Snapshot Execution Register (12329) – allows you to rename a file while it is actively being appended to. Writing a 1 to this register will rename Log001.log to Snap001.log. This will allow a host to upload the Snap001.log file for review while the Log001.log can still be appended in the background.

Snapshot Execution Register (12330) – reflects the results of the operation:

- 0 = SUCCESS
- 53 = ERROR_IOACCESS (log file does not exist or the Snap file already exists)

Data Logging - Log File Deletion

Log Deletion Register (12328) – Writing a file number to this register causes that Log###.log file to be deleted.

Snap Deletion Register (12331) – Writing a file number to this register causes that Snap###.log file to be deleted.

The status of these deletions (0 or 53) are found in:

- Log String Result Register (12327)
- Snap Result Register (12330)

Data Logging - Log Disk Maintenance

Periodically the NVRAM disk will reach maximum capacity. Unlike a disk on your PC when you delete a log file on a NVRAM disk, its space is still consumed on the drive itself. Only its directory reference is destroyed. The ‘format disk’ script command will recover this space. This takes about 300ms for a 4M disk, and needs to be done while no other operations are occurring to the drive.

FlashDiskSelectionRegister (12314) – is used to select the active disk volume.

FlashSpaceAvailRegister (12315) – shows the actual bytes remaining in the active disk volume.

The Script Files

Script files are used to set-up registers and handle the ftp commands. It is assumed you are familiar with basic script commands. Reference the 5200 Script Language Guide and the Advanced Scripting section of the Model 5200 Logging and FTP Client Application Guide for more information.

The following script files include comments that are started with a ‘#’ at the beginning of each comment line. These comments explain what each script does.

The _startup.ini file will run once each time the 5222 is powered up and initializes the controller and creates a virtual disk for data logging.

Script001.ini deletes the log (sets register 12328 =1) It then sets the record to be recorded to the 1st record of the log.ini file to the log file (set register 12325=1). The final line does the write operation (12326 = 1).

Script002.ini does the same thing as Script001.ini except it doesn’t delete the file prior to logging the data. So data is appended to the file.

Script003.ini connects to the FTP server and transfers the log file to the server.

_startup.ini script file

```
#create the virtual disk for data logging  
mount "/_system/Messages/data" "/ramdisk"  
#set the SNTP clock to reset in 60 seconds  
#set to EST time zone; which is -5 hours or -18000 secs from GMT time
```

20043=-18000
20042=60
#12328=1, deletes Log001.log, if exist.
#12325=1, write to Log001.log
#12326=1, file selects data logging for record #1 in the log.ini file. Any additional
#logging will be appended to file.
12328=1
12325=1
12326=1

Script001.ini

12328=1
12325=1
12326=1

Script002.ini

12325=1
12326=1

Script003.ini

*# This script tests the automated transfer of data from
the controller to a remote ftp server.
Register 1 is used as a retry counter
Register 2 is used notify Quickstep program of completion status
Writing to the Script Execution Register (12311) will cause
execution to occur. Script file name is currently Script001.ini
thus writing 1001 will cause this file to run as a thread in the
background, writing a 1 will cause it to run within a Quickstep step.
Threaded execution (background) is the preferred method
#*
:start
*# Clear the retry counter
1 = 0
Clear our completion flag register
2 = -1
goto firstTime*
:retry
*# Delay for 5 seconds and then try again
delay 5000
Bump the retry counter and see if done with retries...
inc R1
if R1 > 3 goto abort*
:firstTime
*# First setup where to branch if an error should occur
onerror goto errorOccurred*

```

# Lets connect to the remote host
    ftpconnect 12.40.53.94 login PassWord

# Now lets upload a file and then download a file
# If an error occurs we will exit automatically due to the 'onerror' command
# SEND to host from controller
    ftpsend /_system/Messages/data/Log001.log
# RECIEVE from host and store to controller under different name
    ftpget anyfile.log /_system/Messages/data/newfile.log
# All done so close host session gracefully
    ftpquit
# Set completion flag to 0 indicating we are done and successful
    2 = 1
# If we get here we are all done
    end
# Process error if should occur
:errorOccurred
# Lets see what type of error occurred
# First check to see if initial connection failed
    if ERRORCODE & 0x00100000 goto retry
# Next see if we failed during a transfer
    if ERRORCODE & 0x00080000 goto retry
# Next see if we failed because the file did not exist on the controller
    if ERRORCODE & 0x00200000 goto nofileController
# Next see if we failed because the file did not exist on the host
    if ERRORCODE & 0x00020000 goto nofileHost
# Was a fatal failure so give up
:abort
    2 = -2
    end
# File did not exist on the controller
:nofileController
    2 = -3
    end
# File did not exist on the host
:nofileHost
    2 = -4
    End

```

Here is the Quickstep program. It monitors the input. Logs the data and runs the script files using the runsript register 12311.

Quickstep Program (Datalog.dsp)

```
[1] Initialize
```

```

    ;;; load current version in NV register
-----
<NO CHANGE IN DIGITAL OUTPUTS>
-----
store 1 to version_r500
goto Next

[2] Start
    ;;; Begin multitasking
-----
<NO CHANGE IN DIGITAL OUTPUTS>
-----
do (DataLog FTP_file_start) goto Start

[3] DataLog
    ;;; Monitor switch #4, when closed, start
    ;;; logging to the controller
-----
<NO CHANGE IN DIGITAL OUTPUTS>
-----
monitor SW4_on goto Delete_log

[4] Start_Log
    ;;; r12325 select the log file to write.
    ;;; r12326 select the record in the log.ini file to write
    ;;; into the log file selected by r12325.
    ;;; Monitor the switch to be deactivated; or
    ;;; after one second, loop back to repeat.
-----
<NO CHANGE IN DIGITAL OUTPUTS>
-----
store 1 to log_execute_r12325
store 2 to log_string_trans_r12326
monitor SW4_off goto DataLog
delay 1 sec goto Start_Log

[5] Delete_log
    ;;; Once switch is activated, the script file
    ;;; Script001.ini will be executed. The script
    ;;; will delete Log001.log and then create a new
    ;;; file. It will then write the header (record 1)
    ;;; to the log file in preparation for data to be
    ;;; written to it.
-----
<NO CHANGE IN DIGITAL OUTPUTS>
-----
store 1 to script_r12311

```



```

delay 1 sec goto Start_Log

[6] FTP_file_start
    ;; This step waits until the switch (3) is activated.
    ;; After which, files will be transfer to and from the
    ;; ftp host to a ftp client
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    monitor SW3_on goto FTP_file_xfer

[7] FTP_file_xfer
    ;; This is where the transfer is commanded. The
    ;; script file (Script003.ini) referred by the
    ;; first statement will be executed.
    ;; It will then wait until the switch is
    ;; deactivated to begin again.
    -----
    <NO CHANGE IN DIGITAL OUTPUTS>
    -----
    store 1003 to script_r12311
    monitor SW3_off goto FTP_file_start

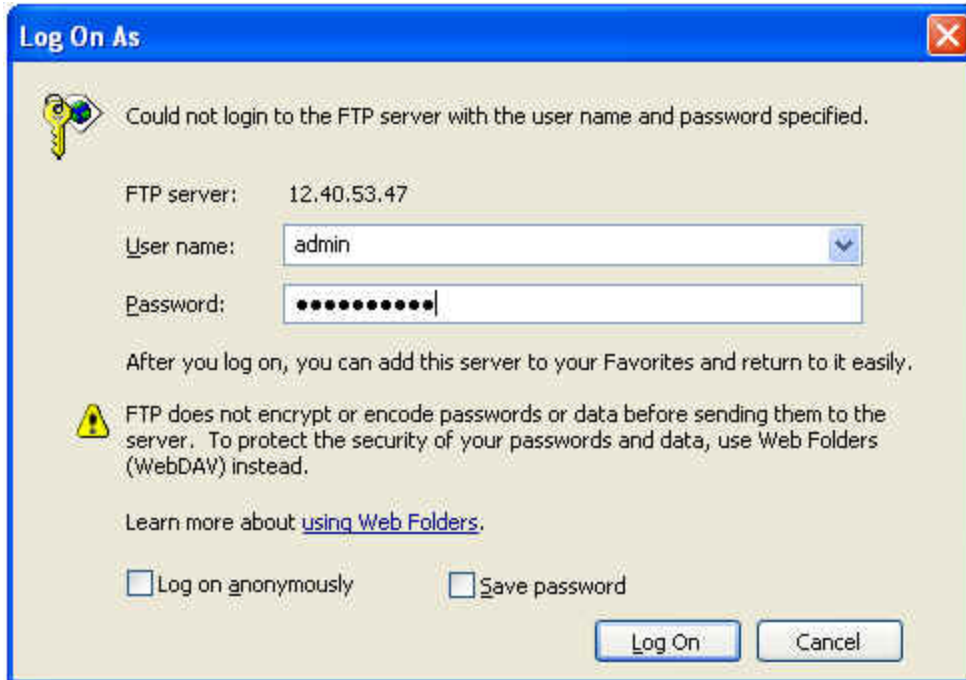
```

To LOAD the script files shown above into the controller:

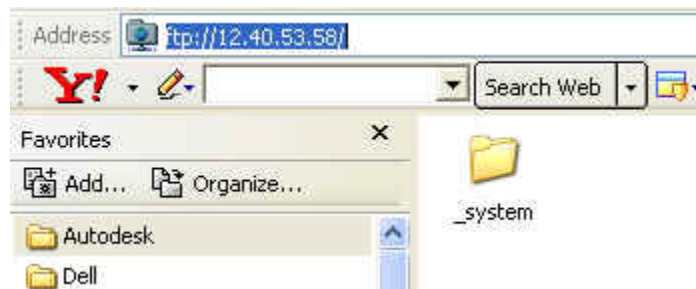
1. Open an Internet Explorer session via ftp



2. Log in to the BlueFusion: username and password is the same as telnet



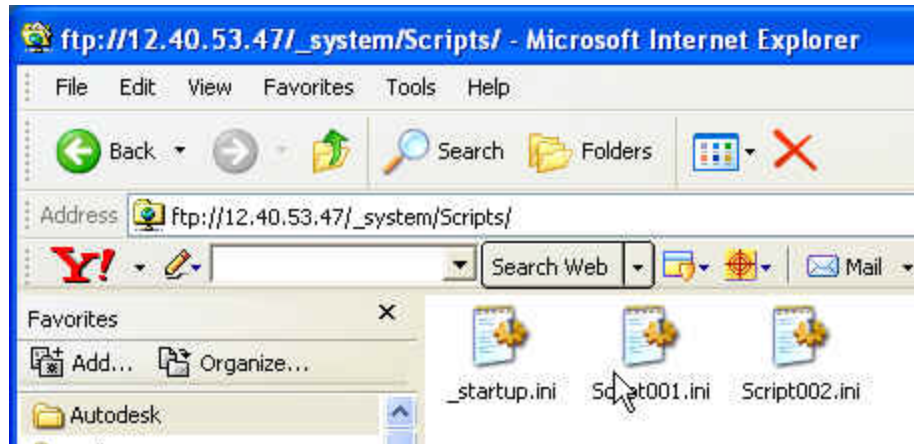
3. Select `_system` directory



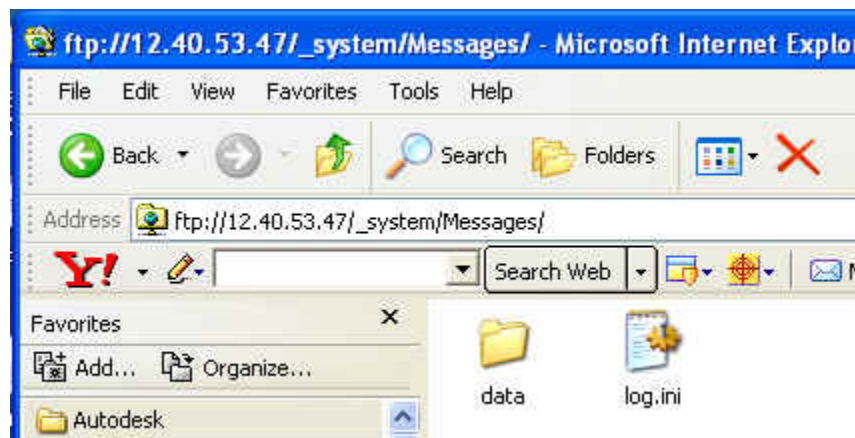
4. The Scripts directory is listed under this directory. Select this directory.



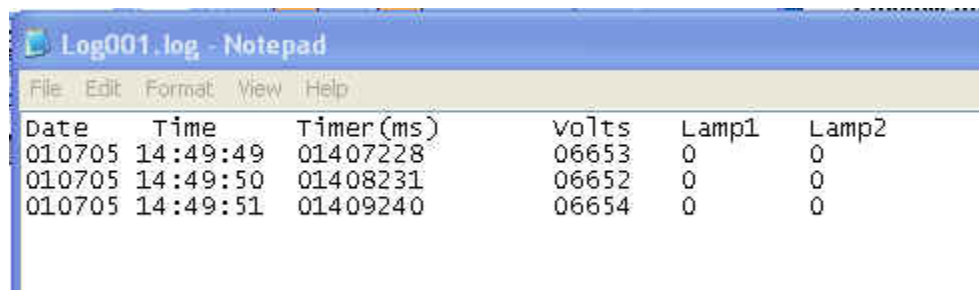
5. Drag and drop the `_startup.ini` script that will be used to mount the virtual directory upon powering up of system in the Scripts directory. (optional)



6. Drag and drop the *log.ini* logging script that will command the data to be logged in the Messages directory.



7. Load the Quickstep program.
8. Run the Quickstep program.
9. Activate switch 4 to begin logging.
10. De-activate switch to stop logging.
11. Open Internet Explorer to drag and drop the log file onto the desktop
12. Open log file with a file like notepad to view contents.



13. Activate the other switch to transfer file via ftp session.

Note: the client ftp site used by the quickstep program is at 12.40.53.94. If a local ftp client is desired, the instructions to install a ftp site is included in the following document starting on page 15: http://www.ctc-control.com/customer/techinfo/docs/5200_951/951-520015.pdf

14. Verify the files were transferred.

```
BlueFusion/_system/Messages/data/>dir
-rw-rw-rw-  0 owner group 000043 JAN 07 15:16 Log001.log
Volume: Capacity - 1581840 Free - 1581120 Deleted - 0000000.

BlueFusion/_system/Messages/data/>dir
-rw-rw-rw-  0 owner group 000043 JAN 07 15:16 Log001.log
-rw-rw-rw-  0 owner group 000041 JAN 07 15:32 newfile.log
Volume: Capacity - 1581840 Free - 1580640 Deleted - 0000000.
```

Before and after transfer

```
BlueFusion/_system/Messages/data/>dir
-rw-rw-rw-  0 owner group 000163 JAN 07 14:49 Log001.log
-rw-rw-rw-  0 owner group 000041 JAN 07 14:49 newfile.log
Volume: Capacity - 1581840 Free - 1567920 Deleted - 0015120.

BlueFusion/_system/Messages/data/>format flash
Formatting disk...
SUCCESS: Disk formatted successfully.

BlueFusion/_system/Messages/data/>dir
Volume: Capacity - 1581840 Free - 1581600 Deleted - 0000000.

BlueFusion/_system/Messages/data/>
```

Note: to change the ftp client that will be used to transfer file to and from the following line need to be changed:

ftpconnect <IP address of client> <username of client> <password of client>

The username & password used can be a user already defined

