PN532 V1.3 March 14, 2001

Windows NT® 4 Device Driver and IEEE CAMAC routines for the Jorway 411S SCSI CAMAC Driver[1] and the Jorway 73A SCSI CAMAC Crate Controller

J. Streets, D. Slimmer

Online Systems Department
Fermilab Computing Division

ABSTRACT

A description of the Windows NT Device Driver, IEEE CAMAC routine interface and Fermilab extensions for the Jorway[2] model 411S SCSI Bus CAMAC Highway Driver and the Jorway model 73A SCSI Bus CAMAC Crate Controller.

® Windows NT is a trademark of the Microsoft Corporation.

[1] The word "Driver", capitalized, refers to the 411S Branch Driver hardware. The word "driver", not capitalized, refers to driver software.

[2] The Jorway Corporation, 27 Bond Street, Westbury, NY 11590.

1. Introduction

The Jorway 411S CAMAC Highway Driver and the Jorway 73A SCSI CAMAC Crate Controller are both supported by the Windows NT SJY v1.1 software product. The driver executable, sjy.sys, is used for both devices. IEEE CAMAC librarys and load time dlls are device specific, but are supported though common source code.

The 411S is the SCSI interface variant of the the Jorway 411 CAMAC Highway Driver. (The UNIBUS and QBUS Jorway 411 Drivers can be upgraded to the 411S Driver.) The software driver and CAMAC standard routines for the 411S and the 73A are available for IRIX and have been used in the DART experiments (ref. Fermilab Computing Division Doc. PN507[3]). The branch Driver has also been run from VMS and ULTRIX.

The Windows NT kernel mode SCSI Jorway driver (SJY) is layered over the NT (SCSI) class driver. The SJYdriver commands are ultimately processed by the miniport driver supplied by the SCSI bus host adapter card manufacturer. SJY driver development was done using the Buslogic BT-946C SCSI bus host adapter and the Adaptec 2940 SCSI bus host adapter installed in a 100 Mhz Pentium IBM PC clone.
The CAMAC library and test routines were written using Micosoft Developer Studio Visual C++ 4.0.

The first release of the SJY driver is complete in that it supports all of the IEEE user functions in single and multi user mode. There are additional features that we would like to add in the future, such as multi-branch support, and Windows 95 support..

Please refer to the Instruction Manual for the 411S SCSI bus CAMAC Highway Driver and the User's Manual for 73A SCSI Bus CAMAC Crate Controller for hardware specific information.

This document describes the SJY installation procedure and the IEEE routines which provide the user interface for CAMAC access. These routines are based on those documented in PN219, "IEEE standard CAMAC Routines".

Readers may be interested in similar software available for IRIX systems - VSD2992 VME-serial CAMAC driver (PN507), CES8210 VME-Parallel CAMAC driver (PN476), RPC-CAMAC driver (PN508), and TCL-CAMAC use interface (PN510).

---

[3] All documents with the label PNnnn or HNmmm refer to Fermilab Computing Division Documents, many of which are available at
http://cddocs.fnal.gov/cfdocs/productsDB/docs.html.

2. Jorway 411S Hardware Installation

The upgrade kit for the Jorway 411S can be installed at the Fermilab Equipment Support Department.
See HN133, “Instructions for modifying the JY411 from QBUS to SCSI Bus” for more details.

2.1 Setting the SCSI ID

The 411S SCSI ID is set with the dip switch on the left hand side of the SCSI interface board. To change the setting, remove the top of the 411S case and locate the DIP switch on the LHS of the top PC board, behind the fan. The SCSI ID of the 411S is set with bits 1(LSB) to 3 (MSB). “ON” is binary 0. Bits 4 (LSB) to 6 (MSB) should be set to the SCSI ID of the PC SCSI host adapter, which is typically 0. Unused SCSI ID’s and the host adapter SCSI ID may often be found by observing the initial PC boot phase where the SCSI ID’s of installed devices are displayed or with utilities provided by the host adapter manufacturer.

2.2 Setting Little Endian

The 411S can be set to swap bytes in hardware using the jumper on the top PC board near the SCSI connector. For IBM PC’s it needs to be set to Little Endian, which is selected by placing the jumper to join the center and left posts, X16 and X17. This is opposite to IRIX systems.

2.3 Power Up

If the 411S is connected to a SCSI bus which contains other SCSI devices, the PC should be shut down so that connecting the SCSI cable to the 411S will not corrupt data on the other devices. Once the 411S is connected to the SCSI bus, both can be powered up. The 411S must be powered up before the computer boots so the PC’s SCSI host adapter can detect the 411S on the SCSI bus. Note that the PC will have to be rebooted after the SJY driver software is installed.

If the 411S is not recognized by the PC’s SCSI host adapter, it is possible that the fuse on the 411S SCSI interface board has blown.

3. Jorway 73A Hardware Installation

3.1 Setting the SCSI ID

The 73A SCSI ID is set with the front panel thumbwheel switch. Select an ID 0-7 that is different from the SCSI host adapter and all other SCSI devices on the same bus. Unused SCSI ID's and the host adapter SCSI ID may often be found by observing the initial PC boot phase where the SCSI ID's of installed devices are displayed, or with utilities provided by the host adapter manufacturer.

3.2 Setting the Host ID

Setting the SCSI host ID is not necessary.

3.3 Setting Little Endian

The 73A can be set to swap bytes in hardware using a jumper located under the module's top cover. For IBM PC's it needs to be set to Little Endian. To set the byte swap order, remove the 73A top cover (has the Jorway data plate). Little Endian is selected by placing the jumper to join the posts labeled X2 and X3. The 73A manual contains a diagram showing the jumper post locations.

3.4 Power Up

If the 73A is connected to a SCSI bus which contains other SCSI devices, the PC should be shut down so that connecting the SCSI cable to the 73A will not corrupt data on the other devices. Once the 73A is connected to the SCSI bus, both can be powered up. The 73A must be powered up before the computer boots so the PC's SCSI host adapter can detect the 73A on the SCSI bus. Note that the PC will have to be rebooted after the SJY driver software is installed.

3.5 Misc.

- SCSI Bus termination: The Jorway 73A is shipped with internal termination resistors installed. If the 73A is not the last device on the SCSI bus, then the internal termination resistors need to be removed. Power for the termination resistors is supplied by the CAMAC crate, so the crate power must be left on to prevent disabling other devices on the SCSI bus.
- Master Crate Controller: When the Jorway is functioning as the master crate controller (not as an auxiliary controller) the front panel REQ should be connected to the front panel G IN.

4. Software Installation

4.1 SJY driver

Only IBM PC's or clones running Windows NT 4.0 are currently supported. A Pentium based system with 100MHz or higher clock speed is recommended.

To install the SJY driver:

1. Log in as administrator. This is necessary because of registry writes. If you attempt to install without being logged in as administrator, the SJY installation hangs in a "Installing components" window. Note that you must also be logged in as administrator to install the SCSI host adapter card, if it is not already installed.
2. Copy the SJY product files to a temporary directory.
3. Double click on setup.exe icon. The setup program will guide you through a brief installation.
4. Reboot the system to start the SJY driver. If not logged in as administrator, and the Jorway is not connected to the SCSI bus and is not powered, the system will not show the "System service failed …" message at boot time. The Jorway must be connected to the SCSI bus and powered for the driver to find the Jorway at boot time.

The setup program copies the SJY driver to the %windir%\system32\drivers directory (unless directed otherwise in setup), and creates appropriate registry entries for the SJY driver. The 411S and 73A dlls are copied to the %windir%\system32 directory. The rest of the the product sources are copied to a directory named C:\CAMAC\Jorway\sjyWIN\…

4.2 CAMAC IEEE Software Library

The IEEE software libraries, sjy411s.lib and sjy73a.lib are located in the bin subdirectory of the SJY driver distribution. The routines contained in the libraries are identical, and are described in detail in following chapters, and in the Computing Division Product Note PN219. The libraries are created for the X86 CPU type, and are compiled with the debug option using Micosoft Developer Studio Visual C++ 4.0.

4.3 Multi-User Support

The SJY driver supports two different processes running the on same PC reading and writing different CAMAC modules through the same CAMAC controller.

4.4 Multi-Branch /Controller Support

Only one Jorway 411S or 73A can be installed on a PC system's SCSI bus. Each 411S, however, can have 2 branches, 1 branch with 62 Serial crates, and 1 branch with 7 parallel crates. True multi-branch support (more than one 411S or 73A per SCSI bus) will be added in the future if there is sufficient demand.

4.5 Examples

The SJY driver distribution contains compiled examples in the bin/example sub-directories. The executables were created using Visual C++, and are supplied in both source and executable format. The examples are:

1. pdt : loops continuously writing an incrementing integer, reading it back, and checking for data integrity.

2. dma: loops continuously writing and reading from a Fermilab Bench Mark Module to test for data integrity. Other CAMAC modules that can respond to an F0 and F16 may also be a suitable substitute for the Fermi Bench Mark Module.

3. qxtab: prints a table of Q and X responses for a particular crate slot.

4. cnaf: executes a single CAMAC command

The example programs may be executed by opening a DOS window and invoking the selected test program. Invoking pdt, dma, and cnaf without arguments will cause the tests to print a argument usage help, and then exit. Qxtab prompts for each argument needed. Note that when running the tests with a Jorway 73A, the branch argument is synonymous with the 73A's SCSI ID. The crate argument is not used, but must be given.

4.5.1 Re-building the examples

Some of the VC++ project build settings may not be saved in the distributed project files sjy411s.mdp and sjy73a.mdp. The following VC++ project settings are necessary to re-build the examples.

1. Select a Default Project Configuration, such as dma-Win32Debug
2. Select Settings … from the top menu bar
3. Select General tab
    Add to Output files … dma (or pdt, qxtab,cnaf)
4. Select Link tab
    Add to Object/library modules ..\..\lib\sjy411s.lib (or \sjy73a.lib)
5. Select the C/C++ tab
    Add to Preprocessor definitions SJY411S (or SJY73A)
    Select Category Preprocessor
        Add to Additional include directories ..\..\inc

There are additional system include files in the distribution. These files are in the sjyWIN\inc directory.

Ntddscsi.h - needed to rebuild the examples
Devioctl.h – needed to rebuild the libraries sjy411s.lib and sjy73a.lib

5. Software de-installation

The Install Wizard installation software is not configured to remove the driver from a system. Manual removal is required.

5.1 Manual sjy de-install

1) Remove CAMAC folder
2) Remove libraries
    %windir%\system32\sjy411s.dll
    %windir%\system32\sjy73a.dll
3) Remove the driver
    %windir%\system32\drivers\sjy.sys
4) Remove the registry entries

Start the registry editor:

a) Click Start
b) Click Run
c) type regedt32 and click OK

Find and remove the sjy registry entries with the Registry Editor:
a) Click the SOFTWARE folder in the HKEY_LOCAL_MACHINE window
b) Click the Fermilab folder
c) Click the sjy sub-folder
d) From the menu bar, click Edit
e) In the Edit menu, click Delete
f) Click yes in the Warning window in response to deleting
   the currently selected key and all of its subkeys

a) Click the SYSTEM folder in the HKEY_LOCAL_MACHINE window
b) Click the CurrentControlSet sub-folder
c) Click the Services sub-folder
d) Click the sjy sub-folder
e) From the menu bar, click Edit
f) In the Edit menu, click Delete
g) Click yes in the Warning window in response to deleting
   the currently selected key and all of its subkeys

6. IEEE CAMAC Standard Library

This function library written to control the Jorway 411S and 73A is based on the European Standards on Nuclear Electronics (ESONE) for CAMAC and the ANSI/IEEE CAMAC standard 758-1979. The library routines use the SCSI kernel mode driver sjy.sys.

All functions have been written in C; all argument types are considered pass-by-value unless they are listed with a "*", in which case they are pass-by-reference. FORTRAN programming stubs have also been written in order to allow each procedure to be called from FORTRAN code, although these were also written in C. Each of these FORTRAN routines have names followed by an underscore (_), and have all arguments passed by reference. The C routines return MURMUR[4] error codes which are defined in the include file camac_murmur_msg_c.h found in the SJY distribution inc sub-directory. The FORTRAN routines are type "void" and do not return values, unless they are functions. A separate call is given to return the error from the last FORTRAN command.

Some of the CAMAC IEEE routines have not been written - most of these concern branch operations, LAM interrupts, and booking. These functions are not currently supported by the software.

6.1 Argument names used in IEEE Standard Routines

The following names are used as arguments in the implementation of the IEEE CAMAC routines.

6.1.1 branch

Jorway 411S: An integer (0-255), indicating the SCSI bus and ID of the Jorway 411S. The branch should be set to 8*SCSI bus + SCSI ID.
Jorway 73A: An integer(0-7), indicating the SCSI ID of the Jorway 73A

6.1.2 cb

A control block, a 4 integer array containing the

1. Requested Repeat Count.
2. Tally (Result repeat count).
3. LAM Identification (not used).
4. DMA type; 0=Q-Stop, 1=Single Word

6.1.3 crate

Jorway 411S: An integer (1-63), indicating the setting of the crate controller.
Jorway 73A: Not used.

6.1.4 *data16

A pointer to the first element of an array of 16 bit integers.

---

[4] DART Message Reporter, document number PN457.

6.1.5 *data24

A pointer to the first element of an array of 32 bit integers.

6.1.6 ext

External address, a 32 bit integer which contains the branch, crate, slot, and sub-address of a CAMAC module.

6.1.7 function

An integer (0-24) to indicate the CAMAC function.

6.1.8 k

Contains the encoded Q and X response as follows:

Value of k	Value of Q	Value of X
0	1	1
1	0	1
2	1	0
3	0	0

6.1.9 lflag

An integer containing 1 for TRUE and 0 for FALSE.

6.1.10 q

An integer containing 1 if there was a Q response, and 0 if there was not a Q response.

6.1.11 slot

An integer (1-30) to indicate the slot address of the CAMAC module.

6.1.12 sub-address

An integer (0-15) to indicate the sub-address of the function for the CAMAC module.

6.2 IEEE Standard Routines

6.2.1 cdreg(*ext, branch, crate, slot, sub-address)

Encodes branch, crate, slot, and sub-address into ext argument.

6.2.2 cgreg(ext, *branch, *crate, *slot, *sub-address)

Decodes ext argument into branch, crate, slot, and sub-address.

6.2.3 cccc(ext)

Clear the CAMAC dataway pointed to by ext argument.

6.2.4 ccci(ext, lflag)

Set (lflag=1) or clear (lflag=0) CAMAC crate inhibit.

6.2.5 cccz(ext)

Initialize the CAMAC branch.

6.2.6 ctgl(ext, *lflag)

Test for Graded LAM, returns LAM value in flag.

6.2.7 cclm(ext, lflag)

Enables (lflag=1) or disables (lflag=0) any on the branch defined by the ext argument.

6.2.8 cfsa(function, ext, *data24, *q)

Executes a single 24 bit CAMAC function.

6.2.9 cfmad(function, ext, *data24, *cb)

Execute a number of 24 bit CAMAC functions, while incrementing the sub-address when there is a Q response or incrementing the slot when there is no Q response.

6.2.10 cfubc(function, ext, *data24, *cb)

Execute a number of 24 bit CAMAC functions at the same address, until there is no Q response. If the last word in cb is 0, the word read when Q is lost is not transferred (Q-Stop). If the last word in argument cb is 1, then the word read when Q is lost is transferred (Single Word). These modes are further described in the IEEE Standard.

6.3 IEEE System-Dependent Subroutines

These subroutines are not part of the ANSI/IEEE standard 758-1979, but are listed in its appendix.

6.3.1 ctstat(*k)

Return the status code of the last CAMAC operation.

6.3.2 cdchn(branch, lflag, route)

If the lflag argument is given as 1, it opens up a branch and maps to the SJY driver device. If the lflag argument is given as 0, the open SJY device is closed (from a previous invocation of cdchn with lflag=1). Note that we have followed the PN219 definitions of the arguments, rather than the appendix in ANSI/IEEE standard 758-1979, to conform to existing code.

Jorway 411S: The branch should be set to 8*SCSI bus + SCSI ID, e.g. branch=12 for the Jorway connected to SCSI bus 1, with SCSI ID 4. Note that if a Serial and Parallel interface on the same Jorway have the same branch, one must use the ccctype routine to select the correct interface.

Jorway 73A: The branch should be set to the SCSI ID indicated by the front panel thumbwheel.

The route argument can be given as 0 for non-rpc applications.

6.3.3 cssa(function, ext, *data16, *q)

Execute a 16 bit mode CAMAC function. Note that we use 16 bit words for data, unlike PN219 which uses 32 bit words for a 16 bit single word transfer. Most FORTRAN applications which used the PN219 convention can be ported by changing the cssa calls to cfsa calls.

6.3.4 csmad (function, ext, *data16, *cb)

Execute a number of 16 bit CAMAC functions, while incrementing the sub-address when there is a Q response, or incrementing the slot when there is no Q response.

6.3.5 csubc(function, ext, *data16, *cb)

Execute a number of 16 bit CAMAC functions at the same address, until there is no Q response. If the last word in cb is 0, the word read when Q is lost is not transferred (Q-Stop). If the last word in cb is 1, then the word read when Q is lost is transferred (Single Word). These modes are further described in the IEEE standard.

6.4 Fermilab Extensions to CAMAC Standard Routines

These routines are not in the ANSI/IEEE standard 758-1979, or in its appendix, however they are listed in PN219, and have been found useful in the past.

6.4.1 camerr

Returns MURMUR status from the last FORTRAN call.

6.4.2 cccbyp(ext, lflag)

Set (lflag=1) or clear (lflag=0) SCC bypass. Returns CAM_S_SUCCESS if successful . Returns CAM_S_PARALLEL if branch is parallel.

6.4.3 cccoff(ext, lflag)

Set SCC on-line (lflag=0) or off-line (lflag=1). Returns CAM_S_SUCCESS if successful . Returns CAM_S_PARALLEL if branch is parallel.

6.4.4 cmsg(sname, data, branch)

Prints an error message containing sname, data, and tally information from the last SCSI INQUIRY, the hadware version and the software version. Note the SCSI INQUIRY contains information only after a CAMAC error is detected.

6.4.5 qrespn

Get Q response from the last CAMAC operation.

6.4.6 xrespn

Get X response from the last CAMAC operation.

6.4.7 ccctype(branch, serial, parallel)

Chooses the Serial or Parallel interface for a branch controlled by a Jorway 411S. Only one of Serial or Parallel is allowed (unlike the earlier UNIBUS and QBUS models.) The default is Serial. In the current implementation of the IEEE routines, one would need to call ccctype between CAMAC transfers to switch between the Serial and Parallel interfaces on the same Jorway 411S. Note that the *ext* argument for a CAMAC transfer is identical for the same crate, slot, and sub-address for Serial and Parallel crates on the same branch.

The Jorway 73A is a Type A controller, so the ccctype function should choose parallel.

Example:

ccctype(branch, 1, 0) - chooses serial
ccctype(branch, 0, 1) - chooses parallel

6.4.8 cderr

This routine has no affect; it is included for compatibility with existing IEEE FORTRAN programs.

6.4.9 csuns

This routine has no affect; it is included for compatibility with existing IEEE FORTRAN programs.

7. IEEE Routines not supported

CTCI: Test Dataway Inhibit
CCCD: Enable or Disable Crate Demand
CTCD: Test Crate Demand Enabled
CDLAM: Declare LAM
CCLC: Clear LAM
CTLM: Test LAM
CCLNK: Link LAM to Service Procedure
CFGA: General Multiple Action
CFUBL: LAM-Synchronized Block Transfer
CFUBR: Repeat Mode Block Transfer
CGLAM: Analyze LAM Identifier
CGREG: Analyze Register Identifier

Unsupported Routines in the ANSI/IEEE Appendix
CCINIT: Branch Initialize
CGCHN: Analyze Channel Declaration
CSGA: General Multiple Action
CSUBL: LAM-Synchronized Block Transfer
CSUBR: Repeat Mode Block Tansfer
CDCRT: Define Crate Identifier

8. Timings

The following data rates were measured between a Fermilab (CAMAC) Bench Mark Module and a 100MHz Pentium system using an Adaptec 2940 SCSI host adapter card and the SCSI Jorway 411S. We measured rates for Serial CAMAC using a Kinetic Systems K3952 Type L-2 Serial Controller, and for Parallel CAMAC using a Jorway Model 71B Crate Controller.

Measurements were taken with the "Free" and "Checked" (has debugging symbols) build version of the SJY driver. Measurements were also taken on a system with the Windows NT "Checked" version installed, and on a system with the Windows NT "Free" version installed. All tests were done using the debug version of the sjy411s.lib library. There was no significant improvement in data rates in any combination of "Free" and "Checked" build versions of the Windows NT and the SJY driver tested.

The timing test program called cfubc/csubc in a tight loop, using buffers that ranged in size from 1 to 32768 bytes. Times were used from the C time function. We found data rates could be described by a set-up time and a rate in kilobytes per second. The 0.8 millisecond setup time was independent of the variables used.

	16 bit	**24 bit**
Parallel	952 Kb/s	1,364 Kb/s
Serial	370 Kb/s	545 Kb/s

9. Appendix A: SJY Kernel Mode Driver

The SJY driver was developed using the Microsoft Windows NT Version 4.0 Device Driver Kit (DDK) and the Microsoft Win32 Software Development Kit (SDK). In this release, the SJY driver, sjy.sys is primarily responsible for creating a device. In the Win32 environment, drivers cannot be accessed directly, so an intermediate software layer interacts with the SJY driver on behalf of the IEEE CAMAC functions using DeviceIoControl (IOCTL like) calls. A small subset of the functionality needed to support the IEEE CAMAC routines is contained within the SJY driver in the form of custom built IO control codes. The remainder is handled with Microsoft provided SCSI_PASS_THROUGH and SCSI_PASS_THROUGH_DIRECT IO control codes. Non-data operations use the SCSI_PASS_THROUGH control code, and data (read/write) operations use the SCSI_PASS_THROUGH_DIRECT control code. Data operations do not use the WriteFile or ReadFile functions in this release because of the difficulty in passing the information necessary to create the SCSI control block to the SJY driver without copying user buffers. Please refer to the SDK documentation for more information on the DeviceIoControl function and the DDK documentation for more information on the SCSI_PASS_THROUGH control codes.

9.1 Errors Returned

Since the SJY driver is in a heavily layered environment, and most requests pass through the SJY driver and are serviced at lower levels, most errors are handled by the Win32 environment. The SJY driver does not return any SJY driver specific errors. A CAMAC read or write error that generates a SCSI Check Condition status will *automatically* generate a SCSI Request Sense. The Request Sense buffer is returned as part of the SCSI_PASS_THROUGH_DIRECT_WITH_BUFFER data structure. A subsequent Request Sense call will return “no sense” because the Sense information for the error condition has already been cleared. Whenever an error condition generates an automatic Request Sense operation, the Request Sense buffer is saved and available for decoding by cmsg() from the IEEE library. Cmsg() also performs a SCSI Inquiry to find the Jorway 411S firmware version, and makes a call to the SJY driver to find the SJY driver version in the Windows NT registry.

9.1.1 Read and Write Errors

The following errors are used internally.

- ENODEV: SCSI ok, no Q on transfer
- EFAULT: SCSI ok, no X on transfer
- EIO: SCSI problem

9.1.2 Implementation Notes

The Request Sense buffer returned by an error condition is automatically cleared after a good CAMAC operation.

9.1.3 LAMs

The SJY driver for IRIX does not have LAM interrupts because the SGI SCSI low level driver does not allow the SCSI bus to interrupt the CPU. The missing feature was found in practice to be unimportant since users expected to poll for LAMs.

The SJY driver does not incorporate LAM interrupts at this time. In the future we will look a the SCSI driver interrupt issue under Windows NT.

9.1.4 DMA size

The maximum size for DMA transfers, using csubc or cfubc for example, is 65536 data buffer bytes.

CAMAC transfer length	**Maximum data buffer words (word = 2 bytes)**
24 bit	16384
16 bit	32768

A 24 bit SCSI transfer actually uses 32 bits - one byte is used for padding.

9.2 Known Bugs

9.2.1 Q response

The Jorway 411S returns a SCSI status of SCSISTAT_GOOD if the CAMAC Q=0, and SCSISTAT_CONDITION_MET if the CAMAC Q=1. In one of the lower NT driver layers, scsiport.sys, the SCSISTAT_CONDITION_MET status is not accepted as a success status (it will only accept SCSISTAT_GOOD). When the Jorway returns SCSISTAT_CONDITION_MET, DeviceIoControl returns a status = 0 (fail) and the SCSI status is cleared (set to 0). This affects CAMAC *data* transfers using the DeviceIoControl function with the SCSI_PASS_THROUGH_DIRECT IO control code. The manufacturer suggested work around is to interpret the correct number of data words transferred *and* SCSI status not equal to SCSISTAT_CHECK_CONDITION as a Q=1 response. Although this is sufficient for most uses, a defective CAMAC module which returns data, but not the correct Q response will not be detected.

This problem does not affect CAMAC *non-data* transfers using the DeviceIoControl function with the SCSI_PASS_THROUGH IO control code.

9.2.2 Device claim

When a Windows NT driver is started, it claims devices for exclusive access. This part of the SJY driver has been disabled.

9.3 SJY build, debugging, and test environment

9.3.1 Hardware

9.3.1.1 PC

A 100Mhz INTEL pentium system running the Windows NT free build (no debugging symbols) with an Adaptec AHA-2940 SCSI host adapter was the main development platform. A second pentium system running the Windows NT checked build (has debugging symbols) with a Buslogic BT-946C SCSI bus host adapter was used for driver debugging. The two systems were connected with a kernel debugging type serial cable. Note that the system used for driver debugging should *not* have a SCSI internal disk. Driver bugs and certain WinDbg commands which can cause a driver to crash will cause file system corruption on a SCSI disk (assuming only one SCSI host adapter).

9.3.1.2 CAMAC

A Fermilab Bench Mark Module was used for the both the Jorway 411S and Jorway 73A tests. However, any CAMAC module that is capable of 24 bit write/read operations may be suitable.

9.3.2 Software

The SJY driver was developed using the Microsoft Windows NT Version 4.0 Device Driver Kit (DDK) and the Microsoft Win32 Software Development Kit (SDK). WinDbg from the SDK was used for driver debugging. The CAMAC library and test routines were written using Micosoft Developer Studio Visual C++ 4.0. Installation software was produced with Microsoft's Platform SDK InstallWizard tools.

10. Appendix B: Developer Notes

10.1 Building the SJY product

10.1.1 Version number

The following files must be changed to increase the version number:

1) Change it in this document.

2) C:\CAMAC\jorway\sjyWIN\install\Aftercpy.cpp –
Change the line "Software\\Fermilab\\sjy\\v1.x" that creates a registry entry. Open the project file _inst32.mdp. Add to the Tools – Options – Directories- Include files C:\MSSDK\iwizard\include. Add to the Tools – Options – Directories – Library files C:\MSSDK\iwizard\lib. Then rebuild all.

3) C:\CAMAC\jorway\sjyWIN\lib\sjy_interfaces.c   sjy_version[6] = "v1.x". This is not currently used since the driver gets the version number from the registry, but its nice to be consistent.

10.1.2 This Document

Create the postscript version of the document by printing this file to a file named pn532.ps. Use the print queue HP Laserjet 4Si/4Si MX PS to produce postscript that can be read on UNIX systems. Delete the first three lines in the postscript file that contain "@PJL".

10.1.3 Rebuild libraries

1) Rebuild sjy411s.lib and sjy73a.lib with C:\CAMAC\jorway\sjyWIN\src\lib.dsw. Select the project configuration "lib – Win32 Debug" and then rebuild all.

2) Rebuild examples with: C:\CAMAC\jorway\sjyWIN\example\sjy411s\sjy411s.dsw; select the project configuration "sjy411s – Win32 Debug" and then rebuild all.
C:\CAMAC\jorway\sjyWIN\example\sjy73a\sjy73a.dsw; select the project configuration "sjy73a – Win32 Debug" and then rebuild all.

3) Rebuild driver with DDK checked build.

4) Open the MSSDK Install Wizard and "repack" project files (create diskette images). Note that CVS seems to corrupt Install1.IWP and this document in .doc form. Also, CVS does not commit _instpgm.exe and setup.exe.

5) Zip up files in C:\CAMAC\jorway\sjyWIN\install\cdrom01 into sjyWIN.zip for KITS.

10.1.4 CVS

The main folder must be named to sjyWIN for cvs operations. This is because the IRIX version of sjy is stored in the cvs repository as “sjy”.

10.2 Testing SJY

These are suggestions for testing most of the sjy product.

1. Rename or remove the sjy\ sjyWIN installation from the target system.
2. Install sjyWIN from the sjy.zip file.
3. Rebuild the libraries – to make sure they can be rebuilt.
4. Rebuild the examples for the same reason as above.
5. Try the sjy411s examples on a parallel or serial controller.
6. Try the sjy73a examples on a Jorway 73a.

11. Appendix B: Release Notes

V1_3 – Addresses bugs identified by Rex Hubbard:
o) Modified sjy_notify_camac_data and sjy_notify_camac_nondata to check sense code 0x9 instead of additional sense code 0x80, and sense code 0x04 instead of additional sense code 0x44. Added checks for sense codes 0x02 and 0x05. Both routines now return –2 for an unknown error.
o) Corrected return value for transfer in case of short transfer or no X.
o) Corrected sjy_cmsg() Hardware Version String output.

The version was built with VC++ 6.0.