CLAS-NOTE-90-006

Sample Program Using the Kinetic Systems CAMAC Interface and PAW

ELTON S. SMITH AND SHOUBEN ZHOU
MAY 17, 1990

Contents

List of Figures

1 Introduction

The programs which read data from the prototypes for the drift chamber and time-of-flight scintillator counters use the Kinetic Systems Corporation Model 3922 Crate Controller, and 2922 Q-bus CAMAC Interface to the MicroVax II. Low level FORTRAN callable CAMAC interface routines are provided by the Kinetic System Corporation 6610-1B Software Support Package [1]. [1] CALIB is a shell for an online program which reads ADC and TDC modules from CAMAC and histograms the charge and times made available to a skeleton analysis. The histograms are stored by HBOOK [2] into a global section which may be accessed and manipulated with the Physics Analysis Workstation (PAW) [3]. Additional graphics capabilities are supported with the HPLOT [4] and HIGZ [5] graphics interface utilities. We use the DCL Command Definition Utility to accept interactive commands which allow for rudimentary control of data acquisition and histogram display. CALIB was written to allow application-specific analysis to be made within a FORTRAN programing environment.

2 CALIB

2.1 Getting Started

In order to run the CALIB program, we recommend that the terminal be setup to display both text and graphic screens ("Both planes" for the GraphOn 235 terminal). Then define the logical name "CAL" to be [DEMO.CALIB]. Then set default directory to [DEMO.CALIB]. Then, to begin data acquisition, type in "RUN CALIB". It will take a minute for CALIB software to initialze both CAMAC and PAW interfaces. The software is ready to accept commands when the string 'CALIB>' is displayed on the screen. To initiate data acquisition type "RUN." The list of available commands can be listed with "HELP." Data acquisition is inhibited with the command "STOP." A listing of all commands is given in appendix A.

[1] These routines replace the CAMAC program EASY written by Peter Dunn.

2.2 CALIB Commands

The CALIB program consists of 13 commands which are used for online control, debug and display. Here we describe their general functionality. The syntax of the commands (see Appendices A and B) is prescribed by the VAX/VMS Command Language Interpreter which is used to accept terminal input.

2.2.1 Control

The commands of RUN, STOP, ERASE, EXIT, and RESET are used to enable and disable LAMs, clear the screen, and zero histograms and program counters. For example, to erase the text on the screen type "ERASE TEXT", to erase graphic displays type in "ERASE GRAPHICS," and if you want to erase both text and graphics, type in "ERASE BOTH." The default is "BOTH."

2.2.2 Debug

If you want to output the raw data as it is read from CAMAC, type "DEBUG ON." The CALIB software will type the raw TDC and ADC data. The display will be continued until you type in "DEBUG OFF." If the parameter is omitted, the program will prompt you for it.

2.2.3 Display

Most of the commands are used for display. The PLOT command will display either 1-dimensional or 2-dimensional histograms. Qualifiers to this command allow the histograms to be displayed with various options such as scatter, contour, lego or surface for 2-dimension histograms. The following are some pictures displayed by PLOT commands. For more details see appendix A. A useful command is BROWSE, which plots all defined histograms one at a time. The two commands META and HARDCOPY produce a metafile with plotted histograms which may then be sent to a laser printer. The following steps will produce a hardcopy of histogram 101 and sent by default to CB113.

CALIB>META	Open the metafile to printed.

CALIB>PLOT 101	Output histogram 101 to the screen and to the metafile.
CALIB>HARDCOPY	Print the metafile.

Other display commands are HISTS, STATS and HELP. HISTS will make a list of all defined histograms and STATS outputs a few run statistics to the terminal. HELP lists all CALIB commands.

2.3 Dynamic Access to Histograms

The histograms defined by CALIB are stored in a global section which may be dynamically accessed with PAW. The global section is named "CALIB." To run PAW on CEBAF3, set the terminal type to 72. [2] To map to the global section, type 'HISTOGRAM/HIO/GLOBAL_SECT CALIB'. This sets the current PAW directory to //CALIB. We give an example how to map the global section with PAW and plot four histograms (Identifiers 101-104) on a single page.

```
PAW>  OPTION STAT
PAW>  OPTION DATE
PAW>  HISTOGRAM/HIO/GLOBAL_SECT CALIB
PAW>  META 10 70 [3]
PAW>  ZONE 2 2
PAW>  HISTOGRAM/PLOT 101
PAW>  HISTOGRAM/PLOT 102
PAW>  HISTOGRAM/PLOT 103
PAW>  HISTOGRAM/PLOT 104
```

The output of this sequence of commands shown in Figure 1 is dispalyed on the screen and stored in the file named PAW.METAFILE. If you want to get a hardcopy on queue CB113, type in the command:

PRINT/SETUP=TEKON/NOFL/QUE=CB113 PAW.METAFILE[4]

[2] Currently CEBAF3 and CEBAF2 are running different versions of PAW. The terminal type and metafile number are different. The terminal type for CEBAF2 is 301400.

[3] On CEBAF2 should be META 10 300018.

[4] To print the metafile on queue CB113 on CEBAF2, use the CERN defined ultility MPLOT. The command is MPLOT PAW.METAFILE .8 CB113

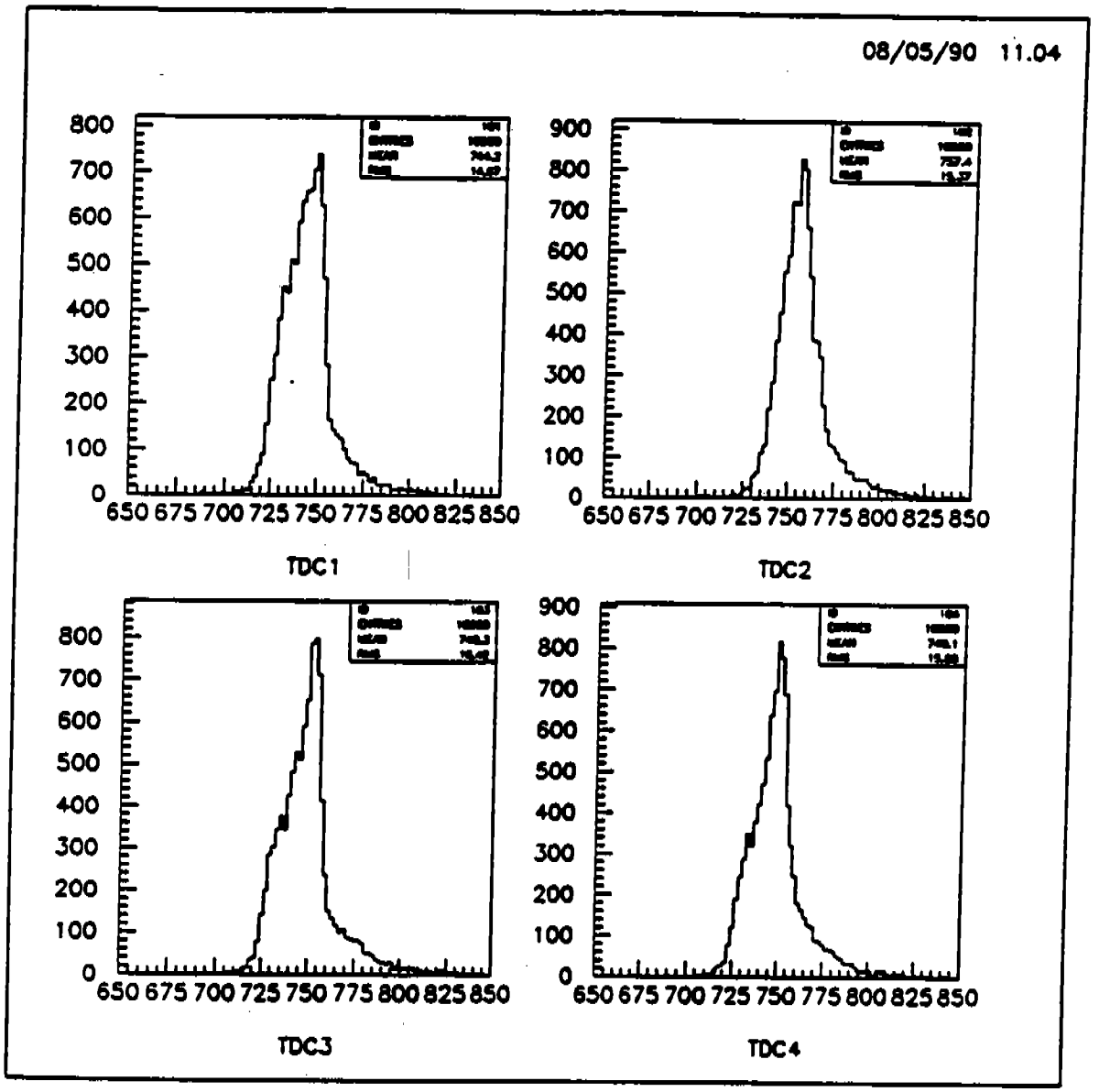


Figure 1: Example of four histograms displayed with PAW.

3 How to modify CALIB

3.1 FORTRAN Source Code

The CALIB software is kept in the public directory DISK$CEBAF3_SYS:[DEMO.CALIB]. The FORTRAN source file is CALIB.FOR, the executable image is CALIB.EXE and the Command Language Definitions are contained in CALIB_COMMAND.CLD. Various parameters are set in the data file CAL_INIT.DAT such as the graphics terminal type, slot number for LAM generation, etc. The command procedure CALIB.COM is used to compile and link the program.

The general flow of CALIB is shown in Figures 2 and 3. The main program waits for command from the terminal and dispatches control to appropriate subroutines which execute the desired function. Data is acquired in parallel by servicing the CAMAC LAMs which have been enabled. The AST routine ANALYZER is called to execute CAMAC operations every event and to perform rudimentary analysis such as filling histograms. Any additional online processing should be added to this routine. CAMAC operations are performed using the Kinetic System Corporation 6610-1B Software Support Package. This package is provided under the current version of VMS and consists of a loadable VMS CAMAC driver and a set of FORTRAN callable interface routines.

3.2 Histograms, Access to CAMAC and Analysis

In general one must modify the subroutines CALIB_PLOT_INIT, and ANALYZER to define additional histograms or to address new CAMAC modules. Histograms are defined at initialization time and filled in the interrupt routine ANALYZER using standard HBOOK calls. At present ANALYZER performs rudimentary analysis in addtion to filling histograms, but its capability may be enhanced as appropriate. CAMAC modules are addressed using FORTRAN callable subroutines which are described in detail in Appendix C. For example, to read 10 channels from slot 5 at crate 0 using a block transer DMA (QSCN mode) use the following call:

```
call cab16(chan,0,5,0,read_code,qscn,buffer,10,status)
```

The crate and LAM number are specified in the data input file CAL_INIT.DAT.

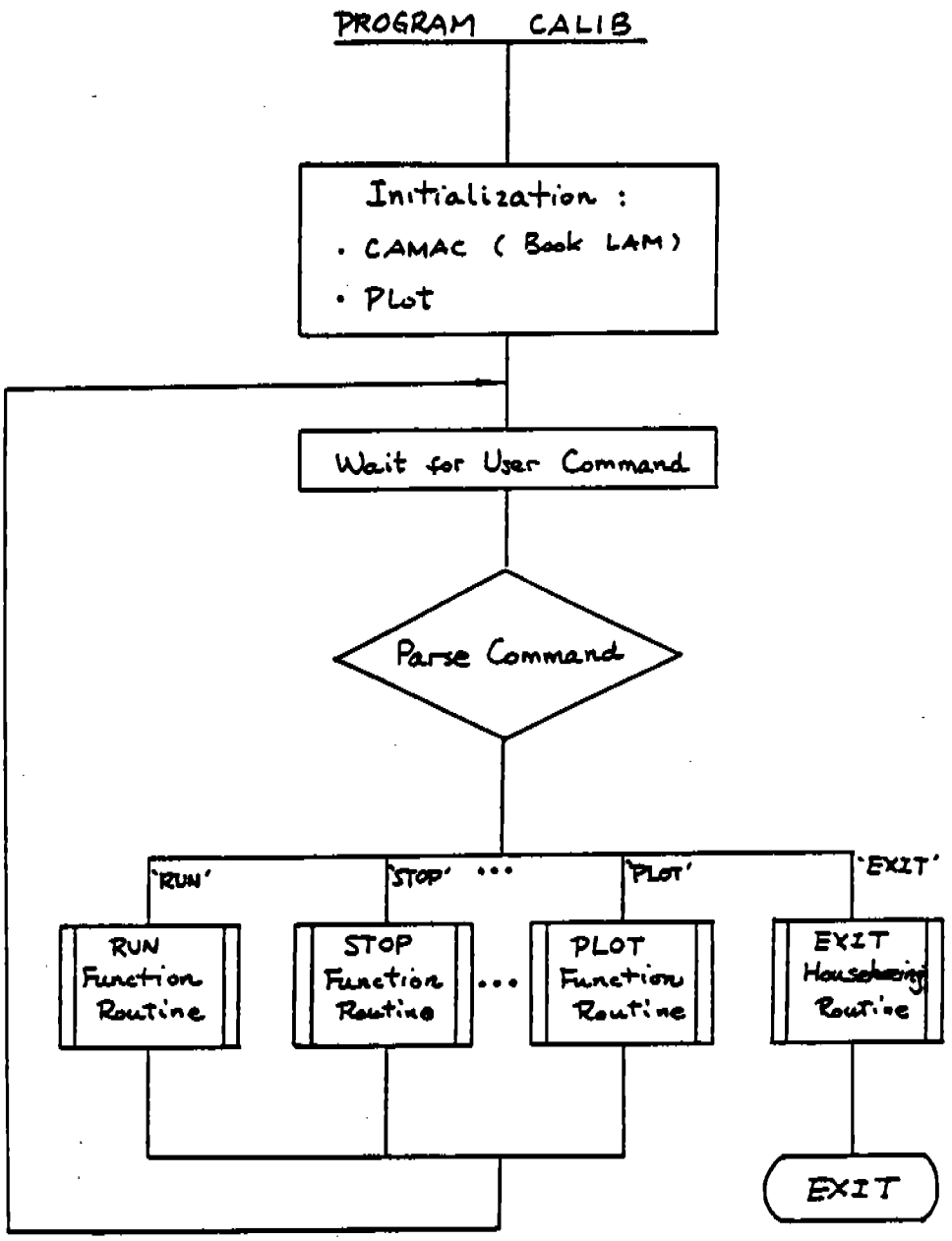


Figure 2: Flowchart of Main Program.

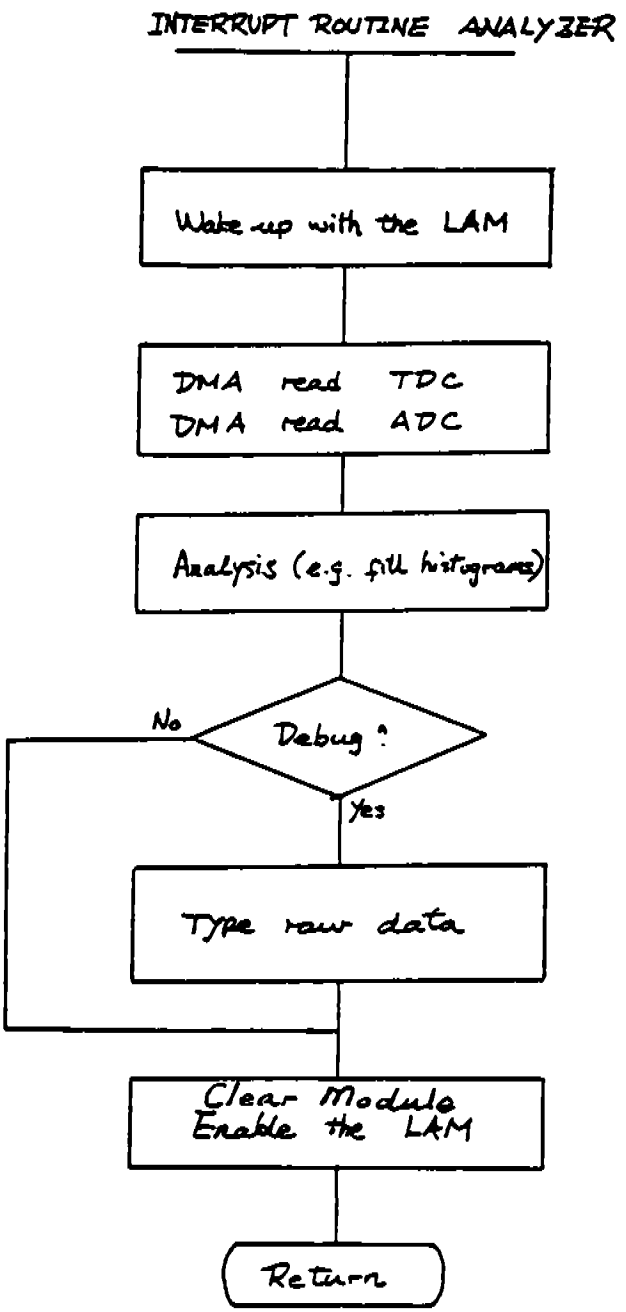


Figure 3: Flowchart of interrupt routine ANALYZER.

A CALIB commands

- BROWSE

  Fast display of all the histograms. Display begins from the histogram's identification number typed in, otherwise from the smallest histogram's identification number. After each display the system will ask the user whether or not he wishes to continue. If not, type 'N' or 'n', otherwise simply type the <CR> key.

  Format :
  BROWSE [histogram-identification-number]

  Parameters :
  Histogram's identification number. Default is the smallest number.

  Qualifier :
  None

- DEBUG

  Enable/Disable DEBUG mode, with Enable(ON) mode, data being read through CAMAC interface will be displayed on the screen, including both TDC and ADC data.

  Format :
  DEBUG mode

  Parameter :
  ON – enable debug mode
  OFF – disable debug mode

  Qualifier :
  None

- ERASE

  Clear the graphics ,text, or both on the screen according to the parameter the user typed it in.

Format :
ERASE parameter

Parameter :
Graphics – Clear the graphics on the screen
Text – Clear the text on the screen
Both – Clear both graphics and text on the screen (Default)

Qualifier :
None

- EXIT

  Exit from CALIB or type CTRL/Z to exit

  format :
  EXIT

  Parameter :
  None

  Qualifier :
  None

- HARDCOPY

  Send metafile to a specific laser printer. The metafile is opened using the command META and plots are made using PLOT.

  Format :
  HARDCOPY[/qualifier] parameter

  Parameter :
  Laser printer device name. The default is CB113.

  Qualifier :
  /(NO)DELETE
  Whether or not to delete meta file after printing. The default is DELETE.

- HELP

  List all defined commands.

  Format :
  HELP

  Parameter :
  None

  Qualifier :
  None

- HISTS

  List of all defined histograms

  Format :
  HISTS

  Parameter :
  None

  Qualifiers :
  None

- META

  Open the metafile to be printed. All subsequent PLOT commands will result in plots displayed on the screen and saved in the metafile.

  Format :
  META parameter

  Parameter :
  File name to be used for metafile. The default name is CALIB.META.

  Qualifier :
  None

- PLOT

This command will display one- or two-dimensional histograms on the screen with various options specified by the command qualifiers.

Format :

PLOT[/qualifiers] Histogram_number

Parameter :

Histogram number "ID"

Qualifiers :

/SCATTER
Plot histogram in scatter mode. SCATTER is a default mode for two dimensional histogram
/LEGO
Plot histogram in lego mode for two dimensional histogram
/CONTOUR
Plot histogram in contour mode for two dimensional histogram
/SURFACE
Plot histogram in surface mode for two dimensional histogram
/LOG
If this qualifier is set, the scale for the Y axis is logarithmic, otherwise it is linear
/STATISTICS
Set this qualifier to output plot statistics.

- RESET

Zero histograms and counters.

Format :

RESET

Parameter :

None

Qualifier :

None

- RUN

Enable interrupts (Look - At - Me)

Format :
RUN

Parameter :
None

Qualifier :
None

- STOP

  Disable interrupts (Look - At - Me)

  Format :
  STOP

  Parameter :
  None

  Qualifier :
  None

- STATS

  List run parameters

  Format :
  STATS

  Paremeter :
  None

  Qualifier :
  None

B VAX/VMS Command Definition Utility

The VAX/VMS operating system has a software tool called Command Definition Utility[6] which may be used to develop command-driven software. The Command Definition Utility may be used to define terminal commands with the same syntax as the Digital Command Language (DCL). The VAX/VMS Command Language Interpreter (CLI) may then be used to parse the user defined commands.

The VAX/VMS CLI routines are used to process command strings using information from a command table. The command table defines parameters and qualifiers for each command. To create or modify a command table, you must write a command definition file and then process this file with the Commmand Definition Utility, see the VAX/VMS *Command Definition Utility Reference Manual*. The CLI routines include:

- CLI$DCL_PARSE
- CLI$DISPATCH
- CLI$GET_VALUE
- CLI$PRESENT

When you use the Command Definition Utility to add a new command to your process command table or to the DCL command table, use the CLI$PRESENT and CLI$GET_VALUE routines in the program that is invoked by the new command. These routines retrieve information about the command string that invoked the program.

When you use the Command Definition Utility to create an object module containing a command table, and you link this module with a program, you must use all four CLI routines. First, use CLI$DCL_PARSE and CLI$DISPATCH to parse command strings and invoke routines. Then, use CLI$PRESENT and CLI$GET_VALUE within the routines that execute each command.

The Command Definition Utility is very sophisticated, but easy to use. We give an example to explain how to build our own program using the Command Definition Utility to help understanding the command input to CALIB.

B.1 Build your own command definition file

Use the text editor to create a command definition file named EXAMPLE.CLD, which defines two commands, EXAMPLE and EXIT. The routine process_example processes the command EXAMPLE and the routine process_exit is executed when the EXIT command is issued. The command EXAMPLE requires that a parameter be entered after the command name. If it is omitted, the CLI service will issue the "File name:" prompt. Qualifiers to the command line are entered after a backslash ("/"). In the present example the qualifier label is "p2" which can be checked by the command processing routine (see process_example below). The EXIT command has no parameters or qualifiers.

```
module command_table
define verb example
parameter p1,label=file,value(required),prompt="File name"
qualifier,edit,label=p2
routine process_example
define verb exit
routine process_exit
```

B.2 Example

We give an example which will process commands defined by the command definition file above.

```
      program my_program
c
      integer*4 cli$dcl_parse,cli$dispatch,lib$get_input
      integer*4 process_example,process_exit
      character*80 a
      external command_table
c
c     Issue a read to the terminal and wait
c
10    call lib$get_input(a,'MY_SOFTWARE>')
      if(.not.cli$dcl_pares(a,command_table)) goto 10
c
```

```
c       terminal input: process command
c
        call cli$dispatch
        goto 10
        end

c-------------------------------------------------------
        integer function process_example
c
c       Process command EXAMPLE
c
        integer*4 cli$present,cli$get_value
        character*80 a
c
c       Check for qualifier "p2"
c           If present, type parameter
c
        if(.not.cli$present('p2')goto 30
        call cli$get_value('p2',a)
        type 10,a
10      format(4x,'Qualifier is present: ',a)
c
c       Get (required) parameter file and type out
c
30      call cli$get_value('file',a)
        type 20,a
20      format(4x,'File name is ',a)
        return
        end
c-------------------------------------------------------
        integer function process_exit
c
c       Dummy command to exit program
c
        call exit
        end
```

C The Kinetic System Corporation 6610-1B Software Support Package

The Kinetic System Corporation 6610-1B Software Support Package is provided under the current version of VMS and consists of a loadable VMS CAMAC driver and aset of FORTRAN callable interface routines. The FORTRAN routines supplied by KSC can be used in conjunction with VMS FORTRAN. These routines may also be used with any language that supports the VMS calling sequence. The standard FORTRAN routines provide a simple direct set of calls to perform I/O operations to CAMAC. The calls are divided into six groups:

- Initialization
  - CAOPEN(chan,device,StatusArray)
  - CACLOS(chan,StatusArray)
- Single-Action Data Transfer
  - CAM16(chan,C,N,A,F,data,StatusArray)
  - CAM24(chan,C,N,A,F,data,StatusArray)
- Block Transfer
  - CAB16(chan,C,N,A,F,mode,DataArray,TransCount,StatusArray)
  - CAB24(chan,C,N,A,F,mode,DataArray,TransCount,StatusArray)
- Status and Control
  - CACTRL(chan,C,func,StatusArray)
  - CCSTAT(chan,C,CrateStat,StatusArray)
  - CAMSG(StatusArray)
- LAM or Asynchronous Action
  - CALAM(chan,C,LAMid,Type,Prio,ASTadr,ASTpar,ClrN, ClrA,ClrF,DsaN,DsaA,DsaF,StatusArray)

The CAMAC interface routines are called either as a subroutine:

```
CALL camroutine(arguments...),
```

or as an INTEGER*4 function subroutine:

```
INTEGER*4 camroutine,IERROR
.
.
.
IERROR=camroutine(arguments...)
```

where camroutine is one of the CAMAC routines . In the case of the subroutine calls, The error condition is returned in the variable StatusArray and follows the usual VMS convention. A "1" is returned if the operation was successful.

D Installation of the CAMAC driver

The 6610-1B software package contains the following files which are distributed by Kinetic Systems Corporation.

CADRIVER.OBJ	The 2922 CAMAC device driver.
CABUILD.COM	Command file to build the driver.
CALOAD.COM	Command file to load the driver.
CAUSER.INC	An include file defining fortran parameters.
CAMAC.OLB	The fortran CAMAC library.
CAMSHR.EXE	Sharable image library.
CAMERR.MSG	Error message file.
66101B.TXT	Driver definition file.
CAMAC.CLD	DCL verb definition file.
CAM.EXE	CAMAC/DCL interface file.
CACTRL.EXE	CAMAC/DCL interface file.
CCSTAT.EXE	CAMAC/DCL interface file.

We give a list of steps (D.1 – D.9) which should be followed in order to install the Kinetic Systems CAMAC Driver.

D.1 Restoring software distribution kit

example commands:
CREATE/DIRECTORY device:[uic]
MOUNT/FOREIGN tape_device_name:(MUA0:)
BACKUP/REW/LOG MUA0:[*...] device:[uic]
DISMOUNT MUA0:

D.2 Making a back-up copy of the software be created

example commands:
MOUNT/FOR MUA0:
BACKUP/REWIND/LOG device:[uic] MUA0:2922.BAC
DISMOUNT MUA0:

D.3 Changing VMS sysgen parameters

In order for the Kinetic Systems Software package to operate correctly a few SYSGEN parameters must be set on the system as shown below:

```
SET DEFAULT SYS$SYSTEM
RUN SYSGEN
SYSGEN>USE CURRENT
SYSGEN>SET REALTIME SPTS 1024
SYSGEN>WRITE CURRENT
SYSGEN>EXIT
```

D.4 Rebooting system by executing SHUTDOWN.COM

D.5 Linking the driver

To link the Driver execute the following command procedure

```
@CABUILD.COM ( linking the driver)
( $link/notrace/exe=sys$system:cadriver cadriver )
```

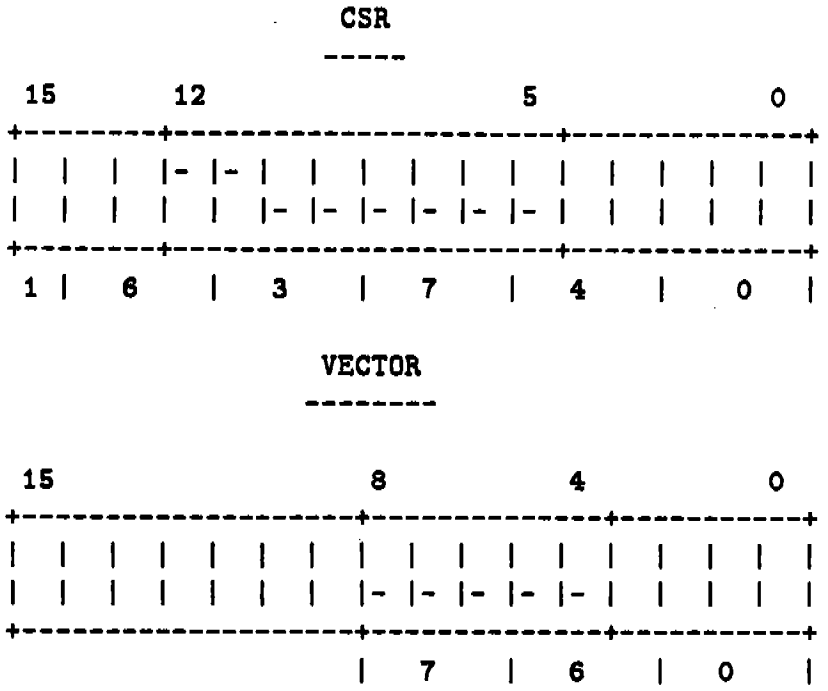
D.6 Checking the connect command in CALOAD.COM

- Before inserting CALOAD.COM into SYSTARTUP.COM, set the switches for the Control and Status Register (CSR) and the interrupt VECTOR address on the 2922 Q-bus CAMAC interface to match the software settings. Care must be taken so that the interface does not conflict with other device settings. We use the factory default settings of

  CSR = 763740
  Vector = 760
  Adapter= UB0

- Checking configuration using the command SHOW/CONFIG with SYSGEN.

- The following switch settings set the CSR and Vector addresses given above



D.7 Checking other commands in CALOAD.COM

Installing CAMAC supporting routines into library and, CAMAC interactive commands into DCL table.

D.8 Changing startup command procedure

Putting changed CALOAD.COM into SYS$MANAGER:SYSTARTUP.COM.

D.9 Rebooting system.

References

[1] "Model 2922 CAMAC VMS Driver 6610-1B Software," Kinetic Systems Corporation, 1989.

[2] R. Brun and D. Lienart, "HBOOK User Guide," CERN Program Library Entry Y250, October 1987.

[3] R. Brun *et.al.*, "PAW – Physics Analysis Workstation, The Complete Reference," CERN Program Library Entry Q121, October 1989.

[4] R. Brun and N. Cremel Somon, "HPLOT User Guide," CERN Program Library Entry Y251, February, 1988.

[5] R. Brun *et.al.*, "HIGZ – High Level Interface to Graphics and Zebra," CERN Program Library Entry Q120, March 1988.

[6] "Command Definition Ultility Reference Manual," Digital Equipment Corporation, 1988.