

# MODAS Users' Manual

David Maden

12<sup>th</sup> August 1994

Stefan Brauchli

30<sup>th</sup> November 1995

Andrea Raselli

12<sup>th</sup> July 2001

This document describes the  $\mu$ SR time-differential ( $\Delta t$ ) data acquisition system.

If you have any suggestions or comments regarding this manual,  
feel free to submit them to

Andrea Raselli  
Paul Scherrer Institut  
WHGA/U150  
CH-5232 Villigen PSI

e-mail [andrea.raselli@psi.ch](mailto:andrea.raselli@psi.ch)  
phone +41(0)56 310 36 10

Document source:	Lmu:\MODAS\Modas_man.doc
Document development:	PC4656 C:\LocalUsers\Raselli\MODAS\Modas_man1.doc
Document printed as:	Acrobat PDFWriter PDF Version 5.0 Document
Last update:	15 <sup>th</sup> April 2004

# Contents

1	Computers and Network Interconnections .....	5
2	General User Notes .....	6
2.1	Cold Starting a MODAS System .....	9
2.2	The "Recover" Procedure .....	11
2.3	Trouble Shooting .....	13
2.4	Shutdown .....	15
2.5	Modus Operandi - Some Hints .....	16
2.6	Hard-copy .....	18
2.7	Securing the VAXstation.....	20
2.8	Disk Quotas and Access to run files .....	21
2.9	Starting and Stopping Runs .....	22
2.10	Looking at Histograms .....	23
2.11	Automatic Run Sequences .....	24
2.12	Session Manager Setup .....	28
2.13	CUPS Printer .....	29
2.14	GPIB Documentation .....	30
2.15	RS232 Documentation .....	32
3	Programs .....	33
3.1	Rigs, Counters, Modes, etc .....	33
3.2	Run Numbers .....	35
3.3	Data File Location .....	36
3.4	Run Control - DELTAT.....	37
3.5	Histogram Display - LOOK.....	42
3.6	The Interpolation Programs .....	48
4	Internals .....	50
4.1	Output Registers .....	50
4.2	Data input to Plotdata .....	50
4.3	HMSR Requests .....	50
4.4	HMSR Data .....	50
4.5	Computer Busy Usage by HMSR.....	51
4.6	SOH and HMSR .....	51
4.7	CAMAC Inhibit.....	51
4.8	STATUS and PTASTATUS Display .....	52
5	Hardware Monitoring.....	54
5.1	Introduction .....	54
5.2	Supported Devices .....	55
5.3	Setup .....	56
5.4	Keithley Setup File Format .....	57
5.5	Conductus Setup File Format.....	59
5.6	AVS-46 Setup .....	60
5.7	Defining Monitoring Limits .....	61
5.8	Keithley Readout .....	62
5.9	Conductus Readout.....	63
5.10	MONITOR Example .....	64
5.11	In Case of Problems .....	66
6	Conductus LTC-20/LTC-21 Temperature Controller .....	68

7 AVS-46 Resistance Bridge & TS520 Temperature Controller..... 69

8 EL734 Step Motor Controller..... 70

9 Facility Instruments at PSI..... 71

10 Ortec pTA 9308..... 72

11 Glossary..... 73

12 Appendix..... 76

13 Index..... 77

## Figures

Figure 1-1 Network Interconnections

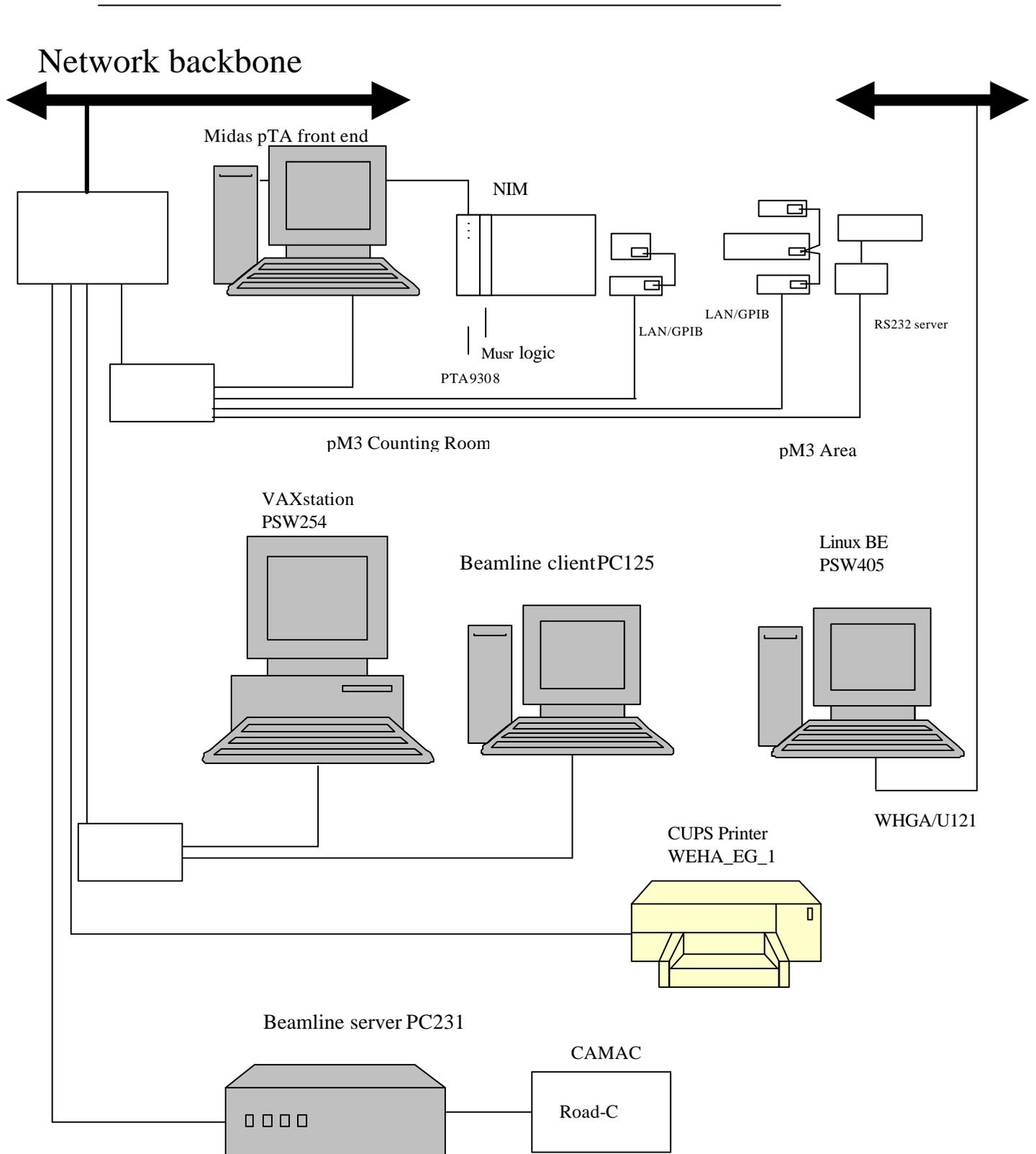
## Tables

Table 2.1 Data Acquisition Computers

Table 2.2 GPIB addresses used

# 1 Computers and Network Interconnections

The overall interconnection of the various hardware items for GPS is shown below, in Figure 1-1.



## 2 General User Notes

Note: *TD\_VAX* refers to the main deltaT (Time-Differential) VAX system for a given experimental area.

*TD\_WS* referred to slave VAXstations that were available in certain areas but are not used anymore.

*FEP* refers to the Front-End Processor. This is either a KAV-30 rtVAX-300 CPU housed in a VME crate and connected to CAMAC via a CBD-8210 VME/CAMAC interface or an ACC-2180 Starburst PDP-11 compatible CPU housed in CAMAC. TDC and HM housed in CAMAC may still be used for data acquisition on the LTF instrument. On the GPD instrument FEP is only used to access slow control devices. On the ALC instrument FEP is used to acquire time integral data.

*BL Server* refers to the secondary beamline TCP/IP server PC, which is connected to the ROAD-C and is located near the beamline power supplies in the gallery.

*BL Client* refers to the secondary beamline TCP/IP client PC located in the counting room or barrack. Programs to access and set the beamline magnets and to optimize the Muon beam are installed here.

*GPIB\_S* refers to the LAN/GPIB gateway (RPC server) being used to communicate with slow control devices that are attached to the same GPIB bus.

*Linux BE* refers to the Linux Midas backend. Midas server and online database are located on this system. Acquired histogram data is now stored on this back end. PSI binary files are then copied to the instrument specific afs directory, where the users may analyze their data.

*MODAS* refers to the Muon Online Data Acquisition System being the main subject of this documentation. MODAS is based on TANDEM components and therefore only runs on OpenVMS/VAX.

*Midas* refers to the Maximum Integrated Data Acquisition System (see <http://midas.psi.ch>). Midas was designed as data acquisition system for medium and small particle physics experiments at PSI and was chosen to replace the TANDEM data acquisition system used at PSI. Midas is based on TCP/IP socket communication and runs under LINUX, UNIX, VMS, VxWorks, Windows NT and MS-DOS operating systems. To prepare the migration of MODAS from TANDEM on VMS/VAX to another operating system new data acquisition equipment used by MODAS (e.g. pTA FE) was developed based on Midas components and Midas run control.

*pTA FE* refers to the Midas Front End; a Windows NT system running the Midas front end application that is designed to operate and read out the Perkin Elmer pTA 9308 picosecond Time Analyzer.

*RS232\_S* refers to the RS232 terminal server being used to communicate with the slow control devices attached to the serial line ports of the server.

*SR Server* refers to the TCP/IP server PC used to remotely set the voltage of the HV power supplies of the spin rotator II. SR server emulates the BL client/server communication protocol. The BL client of the respective area is therefore used to set the

HV voltages.

*TANDEM* refers to Tools for the Acquisition of Data by Experiments and was designed for experiments that use the particle beams of PSI. TANDEM backend software was developed exclusively for the VMS operating system using DECnet and VAX hardware and was never ported to OpenVMS/Alpha or other operating systems. TANDEM front end software was developed for Starburst single board computers using RSX11S as operating system. The CCP (camac command processor) was however ported to the KAV30, which is using the VAXen operating system.

. See <http://ait.web.psi.ch/stdocu/Tandem/tandem.html> to get additional information about TANDEM.

The parameters of the various computers are given in Table 2.1. The definition of other terms may be found in the Glossary.

The procedures described here are not necessarily the only ways of achieving the required goals. In general, one way is described which should work in the  $\mu$ SR environment. If you normally use a different method, feel free to continue doing so. Alternatively, if you have any suggestion or comment regarding the various procedures, you are urged to submit them.

You are reminded that a DECwindows display is considered to be a "personal" tool that can be tailored (customized) to suit your individual style of working. The tailoring operates on a "username" basis. Since you will probably be using a username on the  $\mu$ SR computers which is shared by the other members of your group, you are asked to show due consideration before, for example, "saving" any of the customized settings such as screen colors.

TD\_VAX operates as a standalone VAX/OpenVMS workstation. The current version of the operating system is VAX/OpenVMS V6.2. The display uses the DECwindows/Motif 1.2-3 windowing system.

DECwindows/Motif Operation is not described in this note. Refer to the extracts from the DECwindows User's Guide and the DECterm notes in the MODAS documentation folder. Some familiarity with DECwindows and its terminology (e.g. terms such as "click" and "drag") will be useful to understand the following descriptions.

NOTE: The PSI network migrated into a new network. In the new network each building is a subnet. VAXstations, KAV30s, LAN/GPIB gateways and RS232 terminal server have static IP addresses of the respective subnet assigned. They may not be moved to another subnet without having a new IP address assigned by the network group and being reconfigured for the respective subnet.

**Table 2.1 Data Acquisition Computers and Network Components**

Area	IP Name	Function	Type	DECnet	Internet
GPS $\pi$ M3	PSW254	TD_VAX	VAXstation 4000/90	20.544	129.129.140.236
	PSW405	Linux BE	PowerEdge 2600	-	dhcp
	PC4685	pTA FE	PC	-	dhcp
	PC231	BL Server	PC	-	129.129.140.42
	PC125	BL Client	PC	-	dhcp
	PSGS003	GPIB_S	LAN/GPIB server A	-	129.129.140.40
	PSGS007	GPIB_S 1	LAN/GPIB server C	-	129.129.140.26
	PSTS219	RS232_S	Terminal server A	-	129.129.140.64
GPD $\mu$ E1	PSW255	TD_VAX	VAXstation 4000/90	20.545	129.129.140.15
	PSW404	Linux BE	PowerEdge 2600	-	dhcp
	PC4684	pTA FE	PC	-	dhcp
	PC202	BL Server	PC	-	129.129.140.46
	PC284	BL Client	PC	-	dhcp
	PSGS009	GPIB_S	LAN/GPIB server S	-	129.129.140.249
	PSGS004	GPIB_S 1	LAN/GPIB server A	-	129.129.140.68
	PSTS220	RS232_S	Terminal server A	-	129.129.140.65
ALC $\pi$ E3	PSW214	ALC	VAXstation 4000/90	20.710	129.129.140.18
	PSFE13	FEP	Starburst	20.480	-
	PC2869	pTA FE	PC	-	dhcp
	PC144	BL Server	PC	-	129.129.140.45
	PC278	BL Client	PC	-	dhcp
	PSGS002	GPIB_S	LAN/GPIB server A	-	129.129.140.38
	PSGS010	GPIB_S 1	LAN/GPIB server A	-	129.129.140.248
	PSTS244	RS232_S	Terminal server A	-	129.129.140.67
Dolly $\pi$ E1	PSW251	TD_VAX	VAXstation 4000/90	20.541	129.129.140.17
	PSW403	Linux BE	PowerEdge 2600	-	dhcp
	PC4683	pTA FE	PC	-	dhcp
	PC130	BL Server	PC	-	129.129.140.47
	PC484	BL Client	PC	-	dhcp
	PC1331	SR Server	PC	-	dhcp
	PSGS005	GPIB_S	LAN/GPIB server A	-	129.129.140.69
	PSGS006	GPIB_S 1	LAN/GPIB server C	-	129.129.140.70
PSTS218E	RS232_S	Terminal server A	-	129.129.140.59	
LTF $\pi$ M3	PSW262	TD_VAX	VAXstation 4000/90	20.552	129.129.140.237
	PSFE37	FEP	KAV30	20.504	129.129.140.14
	PSW406	Linux BE	PowerEdge 2600	-	dhcp
	PC4686	pTA FE	PC	-	dhcp
	PSGS011	GPIB_S	LAN/GPIB server B	-	129.129.140.252
	PSGS008	GPIB_S 1	LAN/GPIB server B	-	129.129.140.33
	PSTS243	RS232_S	Terminal server B	-	129.129.140.66

GPIB\_S in the "Function" field denotes the first LAN/GPIB server, whereas GPIB\_S 1 is used for the second LAN/GPIB server. 1000 has to be added to the GPIB address of a GPIB device to assign it to the second server.

LAN/GPIB or Terminal server A, B, C or S in the "Type" field inform about the location (Area, Barrack, Counting room or Studio) of the server.

## 2.1 Cold Starting a MODAS System

This section should only be necessary in extreme cases of trouble such as after a partial or total power failure or replacement of computer hardware. Normally, the *RECOVER* procedure (see Sect 2.2) is a sufficient (and considerably faster) method of getting out of trouble. The following is a brief recipe for booting TD\_VAX and FEP. It assumes, for example, that all cabling is in place and correct. After a significant "beam-off" period, this may not, of course, be the case! This recipe is not intended to be gentle or particularly quick. It is hoped that it will be effective.

1. Locate the various computers associated with the experimental area, viz. TD\_VAX, FEP and pTA FE and establish their type. If still in use, it is particularly important to locate the FEP. For LTF it is a KAV-30 and for ALC a Starburst.

NOTE: Linux BE are located in WHGA/RZ. It should not be necessary to restart Linux BE during a cold start of MODAS unless there is a problem with the Midas ODB database.

2. Turn off<sup>1</sup> TD\_VAX and if still used the crate in which the FEP is located. If there is an Exabyte tape drive connected to the SCSI bus of TD\_VAX, turn it off too. If there are expansion boxes, turn these off too, if they do not automatically get turned off by the power switch on the main CPU box.
3. After turning off any computers or tape drives, wait 20 seconds before turning anything back on again in order to allow time for internal voltages to decay. It is possible that the usual system reset normally associated with a power on is incomplete if this is not done. It is also possible to cause a power-supply to fail by turning it off and on again too quickly.
4. If there is an Exabyte tape drive connected to the SCSI bus of TD\_VAX, turn it on before turning on any computers.
5. If TD\_VAX has any expansion boxes, which are powered separately from the main CPU box, turn them on first. Then turn on TD\_VAX. After a pause it should start executing its power-up diagnostic tests. The text, which appears on the display, will depend on the VAXstation type. It may be a shaded rectangle indicating the progress of the tests.

The computer should then proceed to boot itself. It is possible that the diagnostic tests will detect an error. These cause "?" marks to be displayed together with a diagnostic message. The computer will enter its *console* mode and display the prompt:

```
>>>
```

rather than proceeding to boot. If there are only single question marks displayed<sup>2</sup>, the error can probably be ignored and the boot procedure started manually by issuing the command:

```
>>> boot
```

The boot procedure is lengthy and generates lots of messages on the display, which it is effectively impossible to document. One of the early messages should be of the form:

```
OpenVMS (TM) VAX Version V6.2 Major version id = 1 Minor version id = 0
```

Make sure that MODAS specific startup `PSI_STARTUP_MODAS` and if pTA is used that the Midas specific start up procedure `PSI_STARTUP_MIDAS_MUSR` was executed during start-up.

The entire boot procedure should not, however, take more than 5 minutes before the usual DECwindows login window appears.

6. A message of the form

```
*****
*           TANDEM software now available           *
*****
Hit <Ctrl-F2>, maybe, to get back to normal display
```

<sup>1</sup>If at all possible, it is recommended to shut down the VMS systems on TD\_VAX in a controlled way (see Section 2.4) prior to powering them off.

<sup>2</sup>If an Exabyte drive is connected to the SCSI bus, a "simple question mark" will usually occur in association with Test 6 or Test7. This can be ignored.

should be displayed during the startup procedure. Whatever else you do,

DO NOT LOGIN TO A $\mu$ SR ACCOUNT BEFORE TANDEM IS AVAILABLE!!
---

This is **very** important. If you login too soon, no windows can be created and it may also be impossible to logout again. The computer will need to be rebooted!

7. The 'TANDEM software now available' message will usually be hidden by the DECwindows login window. To get the message to disappear in case TANDEM is available after DECwindows, hold down the  $\langle Ctrl \rangle$  key and press the  $\langle F2 \rangle$ <sup>3</sup> key.
8. If still in use, turn on the FEP. If DECnet is up and running on TD\_VAX and extensive memory tests of the KAV-30 are disabled, the KAV-30 should boot within 20 to 30 seconds. The boot of a KAV-30 is complete when its 2 hex digits display zero. The completion of the boot of a Starburst is a bit more difficult to ascertain without connecting a terminal to its console port. As a suggestion, try studying the Starburst's LEDs after it has been turned on (or after inserting a 50 $\Omega$  terminator in its *RSTR* socket). With the 50 $\Omega$  terminator inserted, the *MEM* and *INTER* LEDs should be lit. On removing the terminator, just the *MEM* light will be lit and then, after a short pause, the *I/O* and *Q22* lights will start to flicker as each block of memory is transferred over the Ethernet into the Starburst's memory. When complete, only the *MEM* LED will be lit followed by 2 flashes of the *I/O* and *Q22* LEDs (corresponding to the output of 2 messages to the console port). With practice, it is quite easy to recognize this sequence!
9. At this point, one can login to a  $\mu$ SR account on TD\_VAX and, once a DECterm window is available, invoke the deltaT run menu and select a *rig* and data acquisition *mode*.

Note: When last measured (June 1994 on PSW254/PSFE43), the above procedure took just under 3 minutes for the "TANDEM Software now available" message to appear. It then took a further 45 seconds to obtain the DECwindows login display on TD\_VAX. The FEP then took 20 seconds to boot. The first login to the MUSR\_TST account on TD\_VAX took 60 seconds before the DCL prompt appeared. Subsequent logins took 45 seconds.

10. Make sure the corresponding Linux BE is reachable over the network. On the DCL prompt PWxx> enter the command "UCX PING PSWxxx" (e.g. PSW405 for GPS) to verify. If Linux BE is not available contact the instrument scientist. To briefly check whether the Midas server is waiting for connections, enter the command "PTASTATUS" on the DCL prompt.
11. Make sure the corresponding pTA FE is reachable over the network. On the DCL prompt enter the command "UCX PING PCxxxx" (e.g.: PC2869 for GPS) to verify.

If the above procedure does not proceed as described, try again and study the messages produced on the display during the boot procedure. These *may* indicate the nature of the problem. Section 2.3 on Trouble Shooting may also be helpful.

---

<sup>3</sup>The F2 key may be labelled Operator Window or Print Screen.

## 2.2 The "Recover" Procedure

As a crude generalization, it is probably true to say that most cases of the data acquisition system getting stuck arise from lack of experience with the windowing system. The user often gets confused between the windows so that, for example, he is not actually logged-in to the correct computer on his currently active window, has started an auto-run sequence from a window which has got lost (e.g. iconized) or has left processes hanging due to incautious use of <Ctrl-Y>.

For these cases, the *Recover* procedure provides a means of re-initializing the data acquisition system. It is usually as good as, but considerably faster than, the cold-start procedure described in Section 2.1. The user is recommended to try this procedure before resorting to the heavy-handed cold-start procedure. In fact, apart from problems associated with hang-ups of the Exabyte tape drive or serial line communication, it is unlikely that a cold-start will help if the recover procedure doesn't!

**Warning:** The *Recover* procedure will kill off **ALL** processes belonging to members of the  $\mu$ SR data acquisition group on TD\_VAX. This includes anyone who is logged-in via "set host" from elsewhere. It will delete any batch jobs which are running, log-out any users of the MUSR user group and delete any windows on the DECwindows displays (except the one from which *Recover* is started). All group logical names will also be deleted. **Therefore first check in section 2.3 if the problem may be solved without recover.**

1. If at all possible, before starting the *Recover* procedure, make a note of the currently selected *rig* and *mode* by checking the heading of the deltaT run control menu. (Type `td` in response to the DCL prompt `$ or PWxxx>` to display the run control menu.) *Rig* and *mode* information will be needed when starting up MODAS again after recovery is complete. If the *mode* indicates "*modified*", try to save the mode settings via the menu item:

```
Modify ...
(TDC) Mode ...
Create/Save Mode
```

2. To start *Recover*, type the command:

```
@ed0:[com]recover
```

in response to the DCL prompt (`$ or PWxxx>`). Then answer the questions carefully.

3. At the end of the procedure, a message of the form:

```
Logging out now to complete the recovery procedure.
Log in again to set up your correct environment.
```

will be obtained. At this point, all windows should have disappeared except for the Session Manager window(s). Select the "End Session" menu option on TD\_VAX to return to the DECwindows login display.

4. NOTE: The MidasD daemon process converting pTA dump files to PSI format and all Mserver processes accessing the Midas ODB database will not be reset during the recover procedure. To reset the ODB database for ALC on the OpenVMS operating system, execute `@ED:[COM]MIDASRC DO ODB RESET` at the DCL prompt (`$ or PWxxx>`). The ODB database for DOLLY, GPD, GPS and LTF on the respective Linux BE has to be reset by the instrument scientist and is therefore not documented here.
5. If FEP is still used

BEFORE LOGGING-IN AGAIN...

reboot the FEP (see step 8. in section 2.1 or Glossary).

6. Log-in on TD\_VAX and start the deltaT run control menu by typing the DCL command:

```
deltat          or          td          at the DCL prompt ($ or PWxxx>)
```

or hitting the <F20> key. You will be automatically prompted to select a *rig* and *mode*.

7. It is usually not necessary to reboot the Midas pTA front end (pTA FE) as the front end application periodically tries to (re)connect to the ODB database after an ODB reset.

## 2.3 Trouble Shooting

What to do in case of problems:

1. Disk Quota Problems. If there are disk quota problems, usually when trying to write the final data file of a run, use the run-control menu item

Extras ...

Show disk quotas

to find out which quota identifier is having trouble. Note that the DCL command "show quota" does not display anything useful. The problem will almost certainly be with *MUSR\_USER* having too many run files in ED:[DLOG] or the user disk being full. Users having disk quota problems with *MUSR\_USER* should be able to relieve the problem, at least temporarily, by deleting old data files from ED0:[DLOG.Dyyyy] or old monitor log files from ED0:[DAT.MONyyyy] and older files in ED:[DLOG] or ED:[DAT].

2. The DECwindows login window does not appear at the completion of booting. Check the license messages during the boot procedure to ensure that important ones have not expired. Take particular note of the licenses for VAX-VMS, DW-MOTIF and DVNETEND.

During start up PSI\_STARTUP\_MOUNT\_USERDISKS, PSI\_STARTUP\_QUEUES, PSI\_STARTUP\_MODAS and if the pTA front end is used PSI\_STARTUP\_MIDAS\_SERVER (on ALC, only) and PSI\_STARTUP\_MIDAS\_MUSR should be executed. These procedures are located in PSI\_COMMON:[SYSMGR] and should be registered in PSI\_STARTUP.DAT.

3. The *TANDEM available* message does not appear at the completion of booting. This can happen if additions to the system have been made since the last boot and the computer now runs out of some resource<sup>4</sup> before the bootstrap completes. There are startup log files of the boot procedures in PSI\_STARTUP that may indicate at which point the startup has failed. The command \$ SHOW LOGICAL /TABLE=SYSTEM\_STATUS may give some information.
4. Lots of errors from MONITOR: This can indicate, that a network connection to FEP or LAN/GPIB gateway is giving trouble, the GPIB communication has got hung up, the GPIB semaphore may be stuck or dead, the GPIB instrument is not configured or temperature sensors are not connected to the temperature controller. Check sensor connections, the setup of the GPIB instruments and the state of the GPIB semaphore. Then check the FEP or LAN/GPIB gateway. Make sure it is powered on. Try to locate and fix the problem as described in section 5.11. Restart the Monitor process and if the problem persists RECOVER.
5. Run Control operation or Modify/Set Device ... is blocked due to waiting for the group logical semaphore to be available. The group logical semaphore is used to avoid access conflicts when devices are configured. It is therefore possible that a device is configured locally or from another terminal or workstation.  
Check all local DECTerm windows to see if setting a device is stuck.  
If this does not help, type \$ SHOW QUEUES to see, whether a batch queue is stopped. To restart a stopped queue, login as user MUSR\_MGR with \$ SET PROC/PRIV=ALL and enter \$ START /QUEUE queue\_name (e.g. PSW087\_BATCH).  
Execute \$ SHOW LOGICAL /GROUP LOGICAL\_SEM to get the PID of the semaphore owner.  
Execute command \$ SHOW USERS /FULL to get information about users being logged on and the PIDs of their processes.  
Execute command \$ SHOW SYSTEM to show a list of all processes, if the PID of the semaphore owner is not found among the user processes.  
Try \$ STOP /ID=pid\_of\_semaphore\_owner to terminate the process, if you can't wait for the semaphore.  
If this does not help RECOVER...
6. If the state of the experiment was changing and the program terminated abnormally or the window was closed during transition, the experiment state has to be reset to state READY. First check that the equivalence name of the logical TD\_STATE is still CHANGING CAMAC\_OK CNTR\_OK MODE\_OK.

---

<sup>4</sup>Global pages is a favourite.

**PW87> SHOW LOGICAL /GROUP TD\_STATE.**

**"TD\_STATE" = "CHANGING CAMAC\_OK CNTR\_OK MODE\_OK" (LNMSGROUP\_000301)**

To reset the experiment state to READY, redefine the logical TD\_STATE

**PW87> DEFINE /GROUP TD\_STATE "READY CAMAC\_OK CNTR\_OK MODE\_OK"**

Data may be lost or it may be necessary to do additional cleanup as the transition might not be completed!

7. If an Automatic Run Sequence is active and the Auto Run window waiting for the required statistics to be accumulated was closed or can't be located.  
Execute \$ SHOW LOGICAL /GROUP DO\_RUNS\_STATUS to get the status (1st field) of the auto run execution and the PID of the auto run execution process (4<sup>th</sup> field).  
Execute \$ SHOW USERS /FULL and \$ SHOW SYSTEM to see if a process with this PID still exists.  
In "Extras .../Auto Run Control ..." select "End Auto Run Sequence" to stop the sequence. When prompted to end the current run enter "YES".  
If the run does get stopped, try "End Run and Proceed" in the "Auto Run Control ..." menu.  
If the run does get stopped either, try \$ STOP /ID=pid\_of\_autorun\_process to terminate the process, if you can't wait for the termination of the auto run.  
Select "Kill Auto Run" to reset the auto run status as the auto run execution process does not exist anymore.  
A currently active run may now be stopped selecting "Terminate run" in the main MODAS menu.  
If this does not help RECOVER...
8. If the problem may not be solved, go to the beamline client personal computer and check whether there is any information in <http://people.web.psi.ch/raselli>.

## 2.4 Shutdown

This section describes the controlled shutdown of TD\_VAX. This is recommended before turning off the power to these computers. There is no shutdown procedure for the FEP; it is simply turned off. The Midas pTA front end is shut down like an ordinary Windows NT workstation, selecting the Shutdown... menu item in the Start menu.

1. Terminate the current session, if any, on TD\_VAX and then login using:

Username: SHUTDOWN  
Password: same as MUSR\_TST

2. After some time (20 to 60 seconds, depending on processor type), a FileView window should appear with a list of users currently logged-in to this computer. This is followed by the question:

Which Node?

Answer with the correct DECnet node name of TD\_VAX, e.g. PSW254. If you are not able to enter the node name, use the mouse to move the arrow on top of the question mark (?) and press and release the left mouse button to select the input field.

3. The question:

Reboot or Shutdown?

should then appear. Answer with "S" to shutdown the VAXstation. If you answer with "R", the VAXstation will shutdown and then immediately reboot itself.

**Note:** If you make any mistakes, the window will disappear. The Session Manager must then be used to select the SHUTDOWN menu item in the Applications menu.

4. After a while (30 to 40 seconds.), the message:

SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM

should appear.

5. Now halt TD\_VAX by pressing its *Halt or Reset Button* hidden under a small plastic cover located on the bottom right side of the front panel. The prompt

>>>

should appear on the display. TD\_VAX can now be turned off, if required.

**Note:** The SHUTDOWN account is privileged and not totally safe. As long as only SHUTDOWN is started, only the above shutdown or reboot capability is possible. However, there is nothing to stop a determined user from starting FileView, for example, and creating havoc.

<b>You have been warned!</b>
------------------------------

## 2.5 Modus Operandi - Some Hints

### 2.5.1 Logging In

The username to use is determined by your group affiliation. The following are commonly available:

MUSR_TST	- General user account, originally for test and development purposes.
CRACOW	- University Cracow
DELFT	- Technical University Delft
IMP	- Institut für Mittelenergie Physik
ITAP	- Institut für theoretische und angewandte Physik
KFA	- Kernforschungsanlage Jülich
KONSTANZ	- Universität Konstanz
STUTTGART	- Universität Stuttgart
RAL	- Rutherford Appleton Laboratory
TUBS	- Technische Universität Braunschweig
TUM	- Technische Universität Muenchen
UNIZH	- Universität Zürich, Chemie
ZH_PHYS	- Universität Zürich, Physik

The passwords are not shown here for security reasons. See your group computer contact person.

Logging in takes quite a long time<sup>5</sup>, especially for the first login after a system bootstrap or recover. The exact appearance of the screen after you have logged in is impossible to predict since it can be tailored to suit your own personal taste (or the taste of anyone else in your group who has used the account). Various windows should appear, probably with titles indicating their intended use. When logging into account MUSR\_TST you may select the "MODAS windows" menu item in the "Applications" menu.

In addition, ordinary DECterm windows, which emulate a VT300 terminal, should be available. One (or more) of these windows can be used for a Telnet session to Compaq TRU64 (PSW284, PSW290) to do off-line analysis there.

To start run control from a DECterm window, issue the DCL command:

```
deltat or td          at the $ or PWxxx> prompt.
```

Note 1: To start run control from a DECterm window, the window must be logged-in on TD\_VAX. Command SHOW PROCESS may be used to verify. If necessary telnet to TD\_VAX to log in. The start-up via the Session Manager includes a login to TD\_VAX (which is why it takes so long).

Note 2: DECterm windows support ReGIS graphics protocol rather than the Tektronix 4014 graphics protocol. Any graphics packages must be configured accordingly or use Xwindows. Alternatively, for really obstinate cases of pre-diluvian software, the public domain XTERM program is available which will create a window, which emulates the Tektronix 4014 protocol. This program can be accessed via the DCL commands:

```
$ import xterm
$ spawn/nowait xterm
```

Having imported XTERM, help on the program is available via *help*. No support is provided for XTERM.

---

<sup>5</sup>30 to 60 seconds, maybe.

## 2.5.2 Logging Auto Run Sequences

Since automatic run sequences are intended to be executed unattended, it is desirable to keep a log of the sequence to allow the possibility of checking for errors. Formerly, this was possible by turning on a slave printer connected to a dumb terminal. With the advent of "window" terminals, this is no longer so easy.

It is possible to keep a disk log of an auto-run sequence via the "*set host*" command. On the window from which the auto run sequence is to be started, issue the command:

```
set host 0 /log=auto_run.log      on the $ or PWxxx> DCL prompt.
```

Everything, which appears on the screen, will then also be written to the file AUTO\_RUN.LOG. When the sequence has been completed, one can log-out and the log-file can be printed (or inspected via one's favorite editor).

## 2.6 Hard-copy

Two possible ways of getting hard-copy from the DECwindows display are described. Both methods operate by taking a snapshot of the screen to a disk file, which can subsequently be printed. The first method will work for any graphic image and is easier for the novice user. The second method is suitable only for making a copy of text on a single DECterm window and is rather convoluted. It has, however, the big advantage that the resulting file is a normal ASCII text file, which can be edited. It is, for example, an ideal method of making a copy of error messages, which can then be annotated and mailed to the appropriate person. See 2.13 for more information about printers and their location.

### 2.6.1 Hard-copy via the Session Manager

This method of obtaining a hardcopy of the DECwindows display should be described in the DECwindows/Motif User's Guide. To summarize, the following steps should be followed:

- o Open the *Applications* drop-down menu on the Session Manager window. If the item *Print Screen* is not present, it must be added by using the *Menus ...* item from the *Options* drop-down menu.
- o Select the *Print Screen* item from the menu. A window appears allowing various options to be selected. The following are recommended as a starting point:

Output Format:	PostScript
Print Orientation:	Portrait
Output File Name:	ED0:[DAT]SCREEN.PS
Ribbon or Toner Saver:	Negative image
Time Delay:	5
Output Color:	Black and White
Fit-to-Paper	Reduce Only

If you are intending to copy parts of the screen, which are white text on black background, select *Positive Image*. If the output file name contains logical name references, ensure that the logical names are defined in the group or system logical name table. Ensure that you have write access to the directory. The name *ED0:[DAT]SCREEN.PS* is recommended, since it is the file printed by the *CUPS\_HC* command.

- o Once one's options have been set, they can be saved via the *Options ...* menu. The window can be resized using the mouse so that only the menu headings are visible. Otherwise, one is likely to get a copy of the *Print Screen* window rather than the part of the screen one really wants.
- o To make a hard copy, select the *Capture Portion of Screen* item from the *Print Screen* drop-down menu. A cross appears which can be used to enclose a rectangular portion of the screen by pressing mouse button 1 at one corner and dragging it to the other corner. On releasing the button, one will have 5 seconds (with the above settings) to rearrange windows (e.g. to ionize the *Print Screen* window) before the hard copy is done.
- o The resulting disk file can now be printed by sending it to the default CUPS printer (e.g. *WEHA\_EG\_1* in  $\pi$ M3 or *WEHA\_12\_1* in  $\mu$ E1). The DCL command *@ED:[COM]CUPS\_HC filename* should do this for you.

Note: In principle, it should be possible to send the hard-copy directly to a PostScript printer via the menu item *Print Portion of Screen* rather than using the 2-step procedure of capturing and then printing. In practice, this has not proved to be possible.

### 2.6.2 Hard-copy of ASCII Text

As mentioned, this method is only suitable for the hard-copy of text information in DECterm windows (e.g. the Run Control menu window or any other window where DCL is being used).

- o First obtain input focus in a window with a DCL prompt and start your favorite editor to edit a new file, e.g.

```
$ lse hc.txt
```

- o Now select the window that contains the text to be written and highlight the text to be copied by clicking and dragging <MB1>. Note that a double click to start the drag will highlight complete

words and a triple click will highlight complete lines. Note that text that has scrolled off the visible window may also be highlighted by first bringing it back into the visible region with the right hand scroll bar.

- o Once the desired text has been highlighted, it can be copied to the file by returning input focus to the editor window and pressing <MB2>.
- o On exiting the editor in the usual way, the file will be closed and can then be printed, mailed, etc.

## 2.7 Securing the VAXstation

When leaving the DECwindows display unattended, one should either *pause* or *end* the session. Note that data acquisition will continue once started even if the terminal from which the run was started logs out.

**Warning:** Do not end a session if an automatic run sequence is running on one of the windows. If you do so, you will have problems terminating the sequence. See section 2.3-7 if this happened.

Since the DECwindows session startup is so slow, it is normal to pause rather than quit the session when leaving the display unattended for a short time. To do this, select the *Pause* item from the "Session" drop-down menu of the Session Manager window. To continue the session, the account's password must be entered.

**Note:** In order that it is clear who is currently using the display when it is in a paused state, it is recommended that the message on the pause screen be modified so that it indicates the user of the display. This message is set up via the *Pause Screen ...* item of the *Options* drop-down menu of the Session Manager window. For example, ITAP might use the text:

```
Type ITAP's password to resume the session
```

**Warning:** If you pause a session on a display for which you do not know the password, the computer will have to be rebooted to regain access to the display.

## 2.8 Disk Quotas and Access to run files

### 2.8.1 Quotas and Access on OpenVMS (ALC)

During a beam period, all (time-differential) data files are written to the directory ED0:[DLOG] and are charged to the MUSR\_USER quota, which is quite large.

As a general rule, data files will not be removed from ED0:[DLOG] until the next beam time starts. However **THIS CANNOT BE GUARANTEED**. Each of the  $\mu$ SR group accounts has a modest disk quota of 20'000 blocks. As data analysis should not be done on the VAXstation this is enough to hold about 150 runs if one has 16k channels of histogram data in each run. Note that charging of data files to your disk quota can be defeated by "renaming" the data files into your directory rather than by "copying" them. Hence, a general policy will be exercised in which it will be assumed that any files in your directory tree which are found to be owned by MUSR\_USER rather than by you may be unconditionally deleted once your beam period has ended.

These rules are in force since the user disk space is a shared resource which, if abused by one or more rogue groups, can adversely affect users who are being more public spirited. As long as disk space is in abundance, the rules are unlikely to be enforced rigidly but no promises can be made. If anyone has suggestions for a better or fairer way of administering disk space, a contribution would be welcomed.

If a file has been deleted on the user disk it may still be available on the local backup disk. Check contents of directory and sub directories of UB0:[MUSR.EXP.TD\_MUSR.DLOG].

Every Thursday morning the temperature log files and the data files created during the last week are archived on ARCHIVFTP and copied to PSIFTP where they are online available.

### 2.8.2 Quotas and Access on Linux BE and AFS (Dolly, GPD, GPS, LTF)

During a run data files are converted to PSI binary format (extension .bin) on the respective Linux BE and then immediately copied to /afs/psi.ch/project/bulkmusr/data/'instrument' (dolly, gpd, gps or ltf) on the PSI afs file servers. There they may be accessed by the users.

Every Thursday morning the data files created during the last week are archived on ARCHIVFTP and copied to PSIFTP, where they are online available (see <http://lmu.web.psi.ch/news/news.html>).

## 2.9 Starting and Stopping Runs

See the documentation on "Run Control" in section 3.4. To summarize, the simplest method is to issue the DCL command *deltaT*, *td* or hit the <F20> key at the DCL prompt (\$ or PWxxx>) on TD\_VAX.

You should obtain a menu of items that enables you to start and stop runs by selecting items using the arrow and "Enter" keys on the keyboard.

## 2.10 Looking at Histograms

### 2.10.1 Looking at Histograms on OpenVMS (ALC)

See the documentation on "Look" in section 3.5. To summarize, you must be working on one of the DECwindows terminals. You may select *Look* as run control menu item:

```
Show ...  
  Look at histograms
```

You should obtain a menu that enables you to look at raw histogram data, perform FFTs and do simple asymmetry plots. Item selection is via the arrow and "Enter" keys on the keyboard.

### 2.10.2 Looking at Histograms on TRU64 or Linux (Dolly, GPD, GPS, LTF)

How to install and set-up the environment (GraphX) to be able to access and use  $\mu$ SR software on TRU64 or Linux provided by the PSI S $\mu$ S facility is described in <http://lmu.web.psi.ch/facilities/software.html> and <http://lmu.web.psi.ch/news/news.html>. Look\_musr, Mufit, Minfit and other applications may be used to view histograms.

## 2.11 Automatic Run Sequences

### 2.11.1 Introduction

The preparation and execution of auto run sequences is done via the

```
Extras ...
  Auto Run Control ...
```

item of the run control menus. You must therefore be logged-in on TD\_VAX to do the following.

Auto run capabilities have so far been implemented so that sequences of runs can be done in which the following parameters are scanned:

- o Magnetic field via the following devices:
  - Oxford Instruments 5403 superconducting magnet power supply (GPIB) for LTF.
  - Oxford Instruments PSC44M power supply (GPIB) for the LTF Helmholtz coils.
  - Gossen power supply (GPIB).
  - ROAD-C (beam line) power supply. These include the devices *WED*, *WEDL* and *WEP* for GPS in  $\pi M3$ , *WEU* and *WEV* for Dolly in  $\pi E3$ , *HELM* and *TSA* for GPD in  $\mu E1$  and  $\mu E4$  and the general purpose device *ROAD*. ROAD-C devices are driven via a TCP/IP connection to the respective beamline server PC of the area.
- o Temperature via the following devices:
  - Neocera Conductus LTC-20/LTC-21 (GPIB).
- o Angular (Dolly, GPD and GPS) and x position (Dolly and GPS) of the sample via an EL734 stepping motor controller. The EL734 is driven via an RS232 terminal server port.

In addition, the following pseudo-devices are available:

- o HE\_FLOW for setting the helium flow of the cryostat.
- o PID\_LAKE for setting the PID parameters of a Lakeshore controller.
- o HEAT\_LAKE for setting the heater range of a Lakeshore controller.
- o ROAD-C (area server) devices in  $\mu E1$  to remotely position the GPD instrument while the beam is on.
- o WAIT for allowing the sequence to wait for a specified period, e.g. to allow a magnetic field to stabilize.
- o SET\_TITLE for setting the run title (SAMPLE, TEMP, FIELD, ORIENT) or sub title (TITLE). NOTE: Spaces may not be entered directly in the title string. Therefore underscores ( \_ ) will be interpreted as spaces. Two subsequent underscores ( \_\_ ) will be interpreted as a single underscore.
- o DCL for allowing an arbitrary DCL command to be issued. By use of this device, devices, which have not yet been integrated into the auto run control system can be used<sup>6</sup>.
- o ALCRUN to perform automatic time integral data acquisition runs on ALC (PSW214).

To get a list of the currently supported devices and their input parameters, create a new run sequence and read the comments about supported devices.

Each run in an auto run sequence is stopped when the count in a particular scaler reaches a predefined value.

### 2.11.2 Preparing an Auto Run Sequence

To set up an auto run sequence, use the Run Control menus to select the item:

```
Extras...
  Auto Run Control ...
```

---

<sup>6</sup>In this way, the AVS-46 temperature controller can be used.

### Prepare Auto Run Sequence

The procedure should be self-explanatory. The header of the run sequence file contains comments about commands and devices that may be used during auto runs. If an editor is used to prepare the run sequence, the header of the file may be viewed.

**NOTE:** If TDC only is used for data acquisition or TDC and pTA are run in parallel, the program STATUS and CAMAC scaler numbering and statistics will be used to determine when the run may be stopped. If pTA only is used for data acquisition the program PTASTATUS and pTA tag numbering and scaler or histograms statistics will be used for end of run determination. See section 4.8 for more information about STATUS and PTASTATUS.

**NOTE:** For ALCRUN (Performing automatic ALC runs on PSW214) enter a comma separated list of parameters (Integer and Floating point), where

<Arg1> = <I1>,<I2>,<I3>,<I4>[,<I5>,<F6>,<F7>,<I8>]

- <I1> = Prescan field value
- <I2> = Field value of first scan point
- <I3> = Field value of last scan point
- <I4> = Field increment
- <I5> = Total number of scans
- <F6> = Wait time after field change for each point
- <F7> = Wait time at start of scan
- <I8> = Preset counting register value

**NOTE:** I1-I4 are mandatory and I5-I8 optional.

**NOTE:** An optional value is only evaluated if all preceding parameter were also defined.

### 2.11.3 Executing a Sequence

Once an auto run sequence has been prepared, the sequence can be started via the menu item:

```
Extras...
  Auto Run Control ...
    Begin Auto Run Sequence
```

It is advisable to keep a disk log of the sequence as described in Section 2.5.2. It is also advisable to set the window banner and icon labels to something recognizable so that the window can easily be identified. Once an auto run sequence has been started and data acquisition enabled, the current run can only be interrupted directly from this window. Other instances of the run-control menu running in other windows can only set flags to signal that the currently active run has to be stopped or that the sequence has to be stopped at the end of the current run. They cannot actually stop the run in all circumstances (See Section 2.3-7 how to stop).

### 2.11.4 Remotely starting a Sequence

Once an auto run sequence has been prepared remotely, the sequence can be started via the menu item:

```
Extras...
  Auto Run Control ...
    Remote Auto Run Sequence
```

A new DECterm window will be opened on the VAXstation display and the new sequence will be started. The items listed in the Auto Run Control menu give information about the processing of the sequence.

### 2.11.5 Interrupting a Sequence

Assume that the auto run sequence was started from terminal or DECterm window A and that a second terminal or DECterm window B is available. Then, start run control on Terminal B and select the menu item

```
Extras...
  Auto Run Control ...
    End Auto Run Sequence
```

The auto run sequence will then terminate at the end of the current run (i.e. when the required statistics have been accumulated).

To stop the current run prematurely, answer the question “End current run?” on terminal B with yes, hit <Q> on Terminal A or select the menu item

```
Extras...
  Auto Run Control ...
    End Run and Proceed
```

Alternatively, hitting <Ctrl-Y> on Terminal A should achieve the combined goal of terminating the current run and then abandoning the remainder of the sequence. It is, however, rather brutal and may be subject to more bugs than the former method.

NOTE: If the DECterm executing the auto run sequence is not available anymore and the auto run sequence may therefore not be stopped proceed as described in section 2.3-7.

### 2.11.6 (Remotely) Interrupting a Run during a Sequence

Assume that the auto run sequence was started from terminal or DECterm window A and that a second terminal or DECterm window B is available on a remote terminal. To stop the current run remotely select the menu item

```
Extras...
  Auto Run Control ...
    End Run and Proceed
```

on terminal B. The current run will then be forced to end and the next run will then be taken from the sequence.

### 2.11.7 Stop waiting for instrument during a Sequence

To be able to remotely stop waiting for an instrument. E.g: to stop waiting to reach a temperature set point on Conductus LTC-21, select the menu item

```
Extras...
  Auto Run Control ...
    Conductus stop waiting
```

The instrument will then stop waiting for the temperature to be stable.

### 2.11.8 Modifying an Auto Run Sequence

If an auto run sequence is in progress, it is possible to prepare a new auto run sequence without interfering with it. Once the new sequence has been prepared, you will be asked if a sequence is currently in progress and, if so, whether you want to switch to the new sequence. If you select to do so, the new sequence will start at the end of the current run.

#### **Warning:**

Any errors occurring during an auto run sequence (e.g. problems encountered setting a new parameter value) will usually result in the run sequence being abandoned with an error message. To prevent loss of beam time:

**Groups using the auto run facility should regularly  
check the status of their experiment**

### **2.11.8 Killing an auto run sequence**

If an auto run sequence was interrupted and the auto run status “CHANGING” is not reset after the currently active run stopped or the run does not stop after reaching the required statistics. Follow instructions in section 2.3-7 to eventually reset the auto run status and stop the run.

## 2.12 Session Manager Setup

The DECwindows Motif Session Manager can be customized so that different “DECterm” windows are created during the session start up. This section describes how to do this. You are expected to have some DECwindows experience to understand the following.

In the Session Manager MenuBar click on Options and select the Menus... item on the Menu to start the Menus window. In the text field located above the DCL Command text field label enter “Start MODAS”. In the text field labeled DCL Command enter “@ud0:[musr.exp.td\_musr.com]start\_modas”. Click on the now appearing upward arrow. The new item will then be inserted into the Item Names field. Make sure Applications is selected in Menu Names. Click on the left arrow to enter the Start MODAS menu item into the Applications menu. Press the OK button to leave the Menus window.

In the Options menu select the Automatic Startup... menu item to start the Automatic Startup window. Scroll in the automatic startup list on the left side and select Start MODAS. Press the OK button to leave the window.

In the Options menu select Save Session Manager to save the modifications.

NOTE: Execute the command @ed:[com]start\_modas to (re)create a “DECterm” that was previously created by Start MODAS and has been closed.

## 2.13 CUPS Printer

For up to date information about CUPS and CUPS printer queues see <http://cups.psi.ch>.

Three laser printers (WEHA\_EG\_1 in  $\pi$ M3, WEHA\_12\_1 in  $\mu$ E1 and WEHA\_E5\_2 below the  $\pi$ E3 area) are currently available in the experimental hall.

Hard copies of monitor or histogram plots in MODAS will be directed to the default CUPS printer. The command `@ED:[COM]CUPS_HC filename` is executed to send the plot file. The symbol CUPSDEST evaluated and printout will be sent to the specified print destination. If you want to send all print files to another print destination, the symbol CUPSDEST has to be (re)defined.

E.G.: `CUPSDEST ::= WLGA_100_1`

If only one (PostScript) file should be sent to another printer you may specify the command

```
$ cups_lp -d <printer_name> <file>
```

The command `HELP CUPS` will give more information about printing using CUPS.

To print ASCII files in VPP style the command file

```
@PSI_PUBLIC:[COM]CUPSPRINT [2UP] [LONG|SHORT] [<destination>] [<filename(s)>]
```

may be executed. 2UP means two pages per sheet (-o number-up=2) and LONG or SHORT means duplex printing flipping on the long or short edge (-o sides=two-sided-long-edge or two-sided-short-edge), respectively. If any of these options is selected text will be printed with 12 characters per inch and 8 lines per inch.

To print ASCII files on the printer without additional formatting, they must first be converted to PostScript (filename.ps) before being printed. A command `CAPS` (`::= PSI_PUBLIC:[EXE]LIST_PS /NOFLAG/LEFT=15/RIGHT=7/CPI=18/LPI=19`) may be defined to perform this conversion. The command format is:

```
$ CAPS <file-spec> "<options>"
```

where <options> can be specified to modify the ASCII/PostScript conversion e.g. the print size and margins can be changed. Help on the available options of the conversion may be obtained via the command:

```
$ help list_ps or $ help caps
```

## 2.14 GPIB Documentation

There are currently two ways to remotely reach GPIB devices. The traditional way is to access GPIB devices via FEP and Kinetics 3388 GPIB interface hosted in the CAMAC crate. On LTF GPIB devices are still accessed using FEP. All other instruments now use one or two LAN/GPIB gateways (RPC server) to communicate with the GPIB devices. See Table 2.1 for name and location of LAN/GPIB devices.

The group logical name GPIB\_SERVER is used to define the first LAN/GPIB server and GPIB\_SERVER1 to define the second server. For the second server 1000 has to be added to the GPIB address of the device to map the device to the second server.

Group logicals for LAN/GPIB servers are defined in the instrument specific section of ED:[COM]LOGIN.COM. New definitions and modifications of existing definitions in LOGIN.COM are exclusively performed by the instrument scientist! To view the currently used definition enter \$ SHOW LOG /GR GPIB\_SERVER\*.

To add or configure a GPIB instrument start the GPIB and RS232 setup procedure:

Extras ...

GPIB & RS232C Setup ...

**NOTE:** The CAMAC station number of the 3388 GPIB interface has to be set to 0. To avoid interference with GPIB\_SERVER. To verify enter \$ SHOW LOG /GR GPIB\_N.

**NOTE:** For the LTC instrument GPIB\_N is set to 15 and the LAN/GPIB server is defined as second server (GPIB\_SERVER1).

The following GPIB addresses are in use on the  $\mu$ SR facility instruments:

Address	Device	Description
0	Kinetics 3388	The CAMAC/GPIB interface
1		
2		
3		
4	Knick or Agilent	DC Power supply to set Helium flow
5	Bruker	Stuttgart magnet power supply
6	SM120-25	LTF Helmholtz power supply
7	Ox. Inst. 5403	LTF superconducting magnet power supply
8		
9		
10	Gossen	Gossen Programmer Power Supply
11	Knick	Knick Programmer Power Supply
12		
13	Conductus LTC-11, LTC-2x	Conductus Temperature Controller
14	Conductus LTC-11, LTC-2x	second Conductus Temperature Controller
15		
16	Keithley 195A	LTF cryostat parameter measurements
17	Keithley 705	Scanner/Input Scaler for Keithley 195A
18		

19		
20		
21	CEC or Agilent (E2050 or E5810)	CEC ISA interface card or LAN-GPIB gateway
22		
23		
24	Oxford ITC50	Oxford Temperature Controller
25	Oxford IPS120	ALC & LTF superconducting Magnet Controller
26		
27	Keithley 199 DMM	CGR monitor
28	Keithley 199 DMM	Diode monitor
29		
30		
31		

GPIB devices are used for hardware monitoring. The background process MONITOR, monitoring slow control devices, is usually involved when there are problems with GPIB and/or MONITOR devices. Follow description in section 5.11 in case of problems.

## 2.15 RS232 Documentation

The AVS46 temperature controller of the LTF instrument is accessed directly using the TTA2: serial line of the VAXstation PSW262.

EI734 stepping motor controllers are driven via an RS232 terminal server port. For more information see section 8.

Setting RVC200 flow on ALC requires the global symbol RS232S\_RVC200. This symbol is defined during logon, when ED:[COM]LOGIN.COM is executed. Variables in ED:[COM]LOGIN.COM are usually defined by the instrument scientist. For more information see ED:[COM]RVC200\_THINGS.COM.

**NOTE:** The Kinetics 3344 CAMAC module with 4 RS232C channels is not used anymore. All 3344 had been replaced by Lantronix RS232 terminal server. See Table 2.1 ("Function" field = RS232\_S) for names and location.

## 3 Programs

This section describes some of the concepts such as *rig*, *counter* and *mode* which are used in MODAS and the various programs which are available for on- and off-line processing.

### 3.1 Rigs, Counters, Modes, etc.

The MODAS data acquisition system is centered around the concept of a "rig". A rig consists of a set of counter telescopes (called "counters" for short) which are routed via a NIM/ECL converter to a TDC and from there to a histogram memory. The number of the input to the NIM/ECL converter in the range 1 to 16 determines the histogram number, which the counter will have in the histogram memory. In addition, up to 24 singles scalers may be included in the system and their contents will be saved in the data file. Scaler overflows are handled in software so that the 24-bit hardware scalers are expanded to 32 bits.

NIM output levels from an input/output register (type IO506 or OR2088) can be used to select or veto the signals from the various counters to the TDC, thus permitting the software selection of which counters are to be histogrammed during a data acquisition run (this automatic feature is not yet in use at PSI). Other outputs from this register may be used to:

- o send a reset pulse at start of run;
- o switch a relay for altering the control sensor input to a KS4580 Philips temperature controller;
- o send a signal from MONITOR that one of its monitored values is out of range or
- o control the diagnostic rate scalers (see the Section on RATE);

The currently used LeCroy TDCs are not manufactured anymore and there is no equivalent replacement available. To replace the currently used hardware (CAMAC Scalors, LeCroy TDC and CES HM) a new  $\mu$ SR data acquisition system based on the Midas software package using the Perkin & Elmer (former EG&G Ortec) picosecond Time analyzer pTA9308 was developed. The new data acquisition system is integrated into the currently used MODAS system and Midas data acquisition is driven by the MODAS run control menu. When performing the rig set-up it is now possible to select TDC and/or pTA for data acquisition. pTA however is not fully integrated into the MODAS system. Therefore it is necessary to perform the set-up of each data acquisition system separately if TDC and pTA are run in parallel.

The parameters of the data acquisition system, which may be set by software, are referred to as the data acquisition "mode". The mode parameters include:

- o the selection of which counters are to be histogrammed. Only "selected" counters are written to the data file.
- o the time resolution of the TDC and/or pTA target format;
- o the number of bins in each histogram (all histograms have the same size);
- o the bin numbers of the time-zero, first-good and last-good bins of each histogram.

"Mode" settings can be saved by name (e.g. "UD2.5" could be used to save settings in which the Up and Down counters are selected and the TDC resolution is set to 2.5 nsec/bin). Up to 10 named TDC modes and 6 named pTA modes are possible for a rig. The run control menus provide various options regarding the TDC and pTA mode setting such as:

- o the loading of one of the preset modes (i.e. setting the TDC, etc. to the values saved in the mode);
- o the modification of the current mode settings;
- o the deletion of a preset mode;
- o the renaming of the current mode and
- o the saving of the current mode settings as a named mode (either superseding the definition of a particular mode or creating a new mode).

The rig and mode setup can be inspected via the

Show ...  
Mode settings

menu item of the run control menus.

The rig definition may be changed via the

- Modify ...
- Rig ...
- Define or Check Rig Parameters

menu item of the run control menus. This is a relatively long operation which is not expected to be needed very often.

Mode settings can be modified via the

- Modify ...
- Mode ...
- Modify Mode Settings

menu item of the run control menus. If a run is active, only the time-zero, first-good and last-good bin settings for each of the selected counters may be changed. No changes are permitted to the TDC resolution, histogram size or counter selection.

## 3.2 Run Numbers

The data acquisition system is very strict about imposing run numbers on data acquisition runs. Run numbers are allocated in a sequence. There are 2 sequences of run numbers, which are used depending on the "type" of the data acquisition. The types are:

- o **TEST** This is the default type which is set immediately after rebooting the computer. At PSI, test run numbers are in the sequence 9991 to 9999. After run 9999, the run number will wrap around to 9991 again. At TRIUMF, the sequence for test runs is from 30001 to 30099. *Test* data acquisition type is intended for runs which are used for setting up purposes and which are not, for example, to be archived or entered in the run log book. If one attempts to start a test run and a data file for the new run number is found to exist already, a warning message will be generated that the old data file will be superseded by the new run but it will be possible to proceed with data acquisition if one insists. It is assumed that test type runs are the exception rather than the rule and so a warning will also be displayed when starting a run if the data acquisition type is set to "test".
- o **REAL** REAL data runs are runs which will be entered in the run log book and archived. At PSI, real run numbers are in the sequence 1 to 8999. At TRIUMF, the real run number sequence is beam-line dependent and is reset to the beginning at the start of each year. If the last run number in the sequence is reached, no more runs may be started without manual intervention. If one attempts to start a real run and a data file for the new run number is found to exist already, a severe warning message will be generated and

### **IT WILL NOT BE POSSIBLE TO START A RUN.**

This situation can only be fixed by deleting (or moving) the offending file manually. The MODAS data acquisition system will, under no circumstances, start a run if it thinks that, by doing so, there is a danger of an existing data file being superseded or deleted.

To switch between REAL and TEST type data acquisition, use the run control menu item

```
Modify ...  
Mode ...  
Acqu Type --> xxxx Data
```

This menu item is available even when a run is active. In this case, the current run number is released, a run number in the new sequence is obtained, any data files with the old run number are deleted and a data file with the new run number is written.

As a hint, should the run numbering sequence get confused and no "expert" is available, one must return to DCL and study the data files in ED0:[DLOG], removing or renaming any which are out of sequence. Following this, one must do a *recover* (see Section 2.2). Then, on the first invocation of "deltat" following the recovery, a check of the run number sequence is performed and one is asked to confirm that the next numbers in sequence are correct. If one says that the sequence is incorrect, one is given the chance to recreate the "Run Number Information File" and if so, one must specify the next real and test run numbers to be used in the two sequences.

**Warning:** The overriding of the run number sequence should be a rare occurrence and is error prone. Please read the messages which appear on the screen very carefully and make sure that you answer correctly.

**Note:** This run number policy was introduced following extensive discussions with physicists at TRIUMF. It may well be unsuited to the PSI style of operation. Please feel free to open a discussion if you think it should be modified.

## 3.3 Data File Location

### 3.4.1 Data File Location on OpenVMS (ALC)

The disk file directory structure of MODAS is based on that recommended by *TANDEM*. This is known to be confusing to the VAX/OpenVMS beginner but is really not too bad once one has got used to it. It is based heavily on the VAX/OpenVMS concept of *concealed device*. The most important of these for the MODAS user is ED0:. This is equivalent to the specification:

```
UD0:[MUSR.EXP.TD_MUSR.]1
```

The MODAS (time differential) data files are in the directory ED0:[DLOG] and have names of the form:

```
DELTAT_<rig>_<rno>.DAT and DELTAT_<rig>_<rno>.BIN
```

where <rig> is the name of the selected rig and <rno> is the run number zero-extended on the left to 4 digits. Files with extension DAT are the original PSI format files containing VMS specific record information and VMS floating point format. Files with the extension BIN may be processed on non VMS systems and real numbers are stored in IEEE floating point format. By default, all MODAS data files have read-access for *World* so that it is not necessary to log-in on TD\_VAX to access a MODAS data file. One must, however, specify the full file name. E.g. to access the data on  $\pi$ E3 from PSW214 for run number 345 which was taken using the ALC rig, say, specify the file name:

```
PSW214::UD0:[MUSR.EXP.TD_MUSR.DLOG]DELTAT_ALC_0345.DAT
```

This file can be opened directly from a Fortran program running on PSW214 provided the keyword *readonly* is specified in the Fortran *open* statement. If one omits the *readonly* specification, the Fortran program attempts to open the file with read/write access and this is not permitted without specifying a  $\mu$ SR username and password as part of the filename.

### 3.4.2 Data File Location on Linux BE (Dolly, GPD, GPS and LTF)

All mdu files located in /userdisk0/mem/musr\_tst/incoming are copied to /userdisk0/musr/exp/td\_musr/dat/midas and converted to PSI bin format located in /userdisk0/musr/exp/td\_musr/dlog. bin files are also copied to the respective afs directory /afs/psi.ch/project/bulkmusr/data/'instrument' (e.g: dolly, gpd, gps or ltf). Errors will be logged in midas.log located in /userdisk0/musr/exp/td\_musr/midas. If a run number change is detected by the midasd.pl daemon midas.log and td\_musr\_save\_pswxx.odb will be renamed to midas.log.old and td\_musr\_save\_pswxx.odb.old, respectively.

---

<sup>1</sup>On the ALC computer, PSW214, ED0: is now also defined as UD0:[MUSR.EXP.TD\_MUSR.], even if ALC (time integral) data acquisition is performed. However data file location of time integral data is UD0:[MUSR.EXP.ALC.DAT.CHEM]

## 3.4 Run Control - DELTAT

Run control operations in MODAS are performed via the run control menu. This section describes some of the options available in this tree of menus.

The run control menu is only available on TD\_VAX. It is started up via the commands:

```
deltat , td_musr or td or by hitting the <F20> key
```

at the \$ or PWxxx> DCL prompt. On other systems the run control menu can be obtained by setting host or opening a telnet session to TD\_VAX.

Once started, the program is menu driven. The menu options, which are available depend on the current state of the data acquisition system so, if you can't find some of the following menu items, it may be that the system is in a state which prohibits their display. To make a menu selection, move the highlighted bar to the item you want by using the up and down keyboard arrows and then pressing the <Return> key. Alternatively, the item may be selected by typing the first letter of the menu text (if it is unique).

After a cold start of the VAXstation or a *recover*, you will first be forced through a check of the run number sequence (see above) and will then be forced through the menu items which select a "Rig" and check its hardware setup. If nothing has changed in the hardware configuration, these steps should be straightforward, if verbose. If old data files have been left on disk in the ED0:[DLOG] directory, you may have trouble with the run number checks. If so, get help to sort it out.

The following headings correspond to menu items in the run control menu tree. In general, if a menu item's text end with an ellipsis ("..."), it will go down a menu level. Selecting "Quit" will go back up one or more levels.

**NOTE:** If a menu item is not shown in the menu, then either the device set-up is not done yet, the device is not used in the currently selected rig or the current state of MODAS (e.g. auto run or run active) disallows usage of menu item.

### 3.4.1 Deltat Top Level Menu

**Begin Run** Once the rig has been selected and checked, you can start data acquisition via "Begin Run". The histogram memory and singles scalers will be cleared and data acquisition will be enabled. At this point, the run title information will be set up via a dialogue. Since data acquisition is already active, you can take your time over this. If a message is displayed indicating that *test* mode rather than *real* mode is selected and you really wanted a *real* run number, it is still OK to proceed with setting up the titles. The mode can be switched later via the Modify .../Modes ... menu.

**Terminate run** Data acquisition will be stopped and the histogram memory contents saved on disk. Before the data is saved but after data acquisition is suspended, you will be asked to confirm the run titles. The data is saved in ED0:[DLOG] as DELTAT\_<rig>\_<run number>.DAT for ALC and in /afs/psi.ch/project/bulkmusr/data/ as deltat\_<rig>\_<run number>.bin for dolly, gpd, gps and ltf, where <rig> indicates the currently selected rig and <run number> is the zero-extended run-number.

**Pause data taking** Data acquisition will be suspended.

**Continue data taking** Data acquisition will continue.

**Write data to Disk** will make a copy of the current contents of the histogram memory and scalers on disk. This is, in any case, done every 5 minutes. Note that only "selected" histograms are copied to disk.

**Show ...** Displays the "Show" menu described below.

**Modify ...** Displays the "Modify" menu described below.

<b>Extras ...</b>	Displays the "Extras" menu described below.
<b>Help</b>	selects DCL help on the MODAS system. It is equivalent to the DCL command: \$ help @modas
<b>Quit</b>	Returns control to DCL.

### 3.4.2 Show Menu

**(TDC) Run Status** gives you status information about the current run if the TDC is used for data acquisition. It is updated every 10 seconds. The 2 lines under "Scalers:" in the display refer to the values of the scalers when they were last saved to disk ("S:") and the current values of the scalers ("C:"). The histogram memory and scalers are normally written to disk every 5 minutes so the "S:" line will not change very often.

**pTA Run Status** gives you status information about the current run if the pTA is used for data acquisition, only. It is updated every 10 seconds. The 2 lines under "Scalers:" in the display refer to the total number of events when the histograms were last saved to disk ("S:") and the current total number of events ("C:"). The histogram data is normally written to disk every 5 minutes so the "S:" line will not change very often.

If TDC and pTA are run in parallel, TDC takes precedence. To get status information of the pTA enter the command PTASTATUS.

**(TDC) Scalers** displays the counts in the CAMAC S500A singles scalers every 10 seconds. The first row of values shows the total values of the scalers (32 bits). The second row shows the change in value since the last display. The third row shows the rate of the last interval.

**pTA Scalers** displays the number of events processed by the pTA every 10 seconds. The first row of values shows the total number of events. The second row shows the change in value since the last display. The third row shows the rate of the last interval.

If TDC and pTA are run in parallel, TDC takes precedence. To have scaler information of the pTA enter the command PTASTATUS SCALERS.

**CAMAC configuration** displays the CAMAC hardware setup. To change it, see the "Modify ..." menu.

**(TDC) Mode settings** displays a list of all the counters in the rig and indicates those which are currently selected (i.e. enabled for histogramming and for writing to the data file). The time-zero, first good and last good bin settings for selected histograms are also shown. The histogram size and resolution are displayed.

**pTA Mode settings** displays Midas equipment name, front end name and the currently selected mode. The selected data format, resolution and histogram size are displayed. A list of all the counters which are currently selected including time-zero, first good and last good bin settings are shown.

**Titles** displays the run titles.

**Auto Run status** shows the auto-run status. A list of the sequence of runs is given and, if a sequence is in progress, a pointer indicates the current run.

**Beam settings (ROAD-C)** this item will cause all beam-line devices currently defined in the device list on the beam-line server PC to be read out and displayed. The file ED0:[DAT]DEVICE.ATARI that is updated from the beamline server PC after each reboot or recovery operation is only used by Modify/Set Device .../ Beam ROAD-C Devices.

**Area settings (ROAD-C)** this item will cause all devices currently defined in the device list on the area server PC to be read out and displayed. The file ED0:[DAT]DEVICE.AREA that is updated from the area device server PC after each reboot or recovery operation is only used by Modify/Set Device .../ Area ROAD-C Devices.

**Look at histograms** If no DECwindows display has been defined, you will be asked to specify which workstation you are working on.

### 3.4.3 Modify Menu

**Zero Run** will suspend data acquisition, zero the histogram memory and singles scalers and start data acquisition again. It is only available if a run is active.

**Kill Run** will terminate data acquisition, delete any data files written with the current run number and make the run number available again.

**Set Device ...** will allow you to set various items such as magnetic fields, angle positions or temperatures by computer control. The list of devices naturally depends on what is available. This menu item is only available if a run is not active.

**Rig ...** allows various modifications of the rig parameters to be made, including the selection of a new rig. With the exception, perhaps, of this item, this menu is not likely to be of interest to the normal user. Selection of TDC and/or pTA for data acquisition and concordance checking between MODAS and Midas run states is done in Define or Check Rig Parameters.

**Mode ...** displays the "Mode" menu described below.

**Titles** invokes a dialogue in which the run titles may be changed.

**Customise ...** produces a menu of items which affect the way the run control menu system operates. It provides a way of setting the window label and also provides a "back-door" into the *set device* menu when a run is active.

### 3.4.4 Extras Menu

This menu contains a few miscellaneous items which do not conveniently fit under other topics. These are:

**Diagnostic Rate Scalers ...** Produces a menu related to the setup program, RATE. This is obsolete for pTA data acquisition.

**Auto Run Control ...** Produces a menu for setting up, starting and stopping automatic run sequences. These are described in Chapter 2.11.

**GPIB & RS232C Setup ...** Produces a menu for setting up parameters related to the GPIB interface and the control of equipment connected to the serial line ports of a Lantronix terminal server. Equipment connected via GPIB or RS232 will not be available as menu option if it has not been set-up with this menu.

**KS4580 Setup ...** Produces a menu for setting up and controlling a KS4580 Philips temperature controller which is not used anymore.

**Monitor ...** Produces a menu for setting up, starting and controlling the MONITOR program. Pen recorder display of the currently monitored devices or viewing of previously recorded monitor log files is also possible.

**Midas Setup ...** Starts the MIDASSET program to configure, select and modify pTA mode settings and view the Midas ODB database. This should only be used by the instrument scientist to perform/alter the initial setup of the database.

**WARNING: It is not possible to perform a Midas run transition (e.g. starting data acquisition of the pTA) while MIDASSET is in editing mode!**

**Show disk quotas** Shows information about users and their disk quotas. This is helpful if data may not be written to disk.

### 3.4.5 (TDC) Mode Menu

Menu items are presented without specifying TDC if TDC is used exclusively. If TDC and pTA are run in parallel the TDC mode menu and the pTA mode menu are combined. Only pTA menu items that are essential are then shown in the Mode menu. All other pTA mode options may be chosen when pTA Mode Operation... is selected.

**Load a Preset (TDC) Mode** gives a menu of the defined modes that may be selected.

**Delete a Preset (TDC) Mode ...** similarly gives a menu of the modes that may be deleted. You are not allowed to delete the current mode.

**Create/Save (TDC) Mode ...** allows the mode settings to be saved in the Run Control Database so that they can be recalled at a later date via the "Load a Preset Mode" option. There are 3 possibilities when saving the mode settings, viz.

- i) Overwrite the currently selected mode with the current settings;
- ii) Create a new mode with the current settings;
- iii) Overwrite some other existing mode with the current settings.

Of these, i) is probably the most common.

Note that ii) will not be allowed if 10 mode have already been created. If you create a new mode, the name you specify must begin with a letter or number. Letters may be in upper or lower case according to taste. In cases ii) and iii), the specified mode becomes the current mode.

**Rename Current (TDC) Mode** does what it says.

If you get into trouble with run control, you can usually recover by hitting <Ctrl-Y>.

### 3.4.6 pTA Mode Menu

A mode must be selected before it may be modified. The last selected mode will be used when data acquisition is started.

If TDC and pTA are run in parallel the TDC mode menu and the pTA mode menu are combined. Only pTA menu items that are essential are then shown in the Mode menu. All other pTA mode options may be chosen when pTA Mode Operation... is selected in the Mode menu.

**Show selected pTA Mode Settings** shows the set-up of the currently used mode.

**Modify selected pTA Mode Settings** modifies the set-up of the currently selected mode.

---

<b>Select current pTA Mode</b>	shows a list of defined modes and prompts for the name of the mode to be in use.
<b>Show pTA Mode Settings...</b>	shows a list of defined modes and prompts for the name of the mode to be shown.
<b>Create pTA Mode</b>	creates a new pTA mode. If an instrument specific template file is available (mode_template_WSNAME.odb in /userdisk0/musr/exp/td_musr/midas on Linux BE or in ED:[MIDAS on VAXstation]) mode set-up will be read from the template file. Currently the maximum number of modes is limited to 6.
<b>Copy pTA Mode</b>	does what it says.
<b>Rename pTA Mode</b>	does what it says.
<b>Delete pTA Mode</b>	does what it says.
<b>Save pTA Mode</b>	saves mode into an ASCII file (*.odb) in directory /userdisk0/musr/exp/td_musr/midas on Linux BE or ED:[MIDAS] on VAXstation.
<b>Restore pTA Mode</b>	reads mode information from an ASCII file (*.odb) located in /userdisk0/musr/exp/td_musr/midas on Linux BE or ED:[MIDAS] on VAXstation.
<b>pTA Mode Operation...</b>	shows a second menu where a mode operation may be selected. This menu option is only available if TDC and pTA are run in parallel.

## 3.5 Histogram Display - LOOK

**NOTE: See section 2.10.2 for information about using look\_musr on Linux or TRU64.**

This program displays histogram data from CAMAC or from a file. It is an X Window System based application that may only be run on an X display.

The program is started, when Look at histograms is selected in the Show Menu. The user has to specify the display name and the transport protocol if they are not yet specified.

Once started, the program is menu driven. Histogram displays appear in windows of their own. To make a menu selection, use the up or down arrow to move to the required item and hit <Return>.

### WARNINGS:

1. To change the operation mode of the program, input from the keyboard is required (e.g. when terminating a continuous display and when selecting data display from file rather than from CAMAC). In these cases, the DECterm window **MUST HAVE INPUT FOCUS**. In particular, it must **NOT** have been iconised. Otherwise, it appears as though the program is dead.
2. For LOOK and DELTAT to operate on an X-Display or VAXstation the Customize/Security menu of the Session Manager or the X-display server must have been set up to allow accessing the display. To allow access you have to specify the transport (DECNET or TCPIP), the node name for DECnet or the hostname for TCPIP. For TCPIP transport the hostname including domain name should be specified (e.g.: psw109.psi.ch). To avoid problems (especially when using TCPIP transport) always enter \* as user name.

The program setup parameters are obtained from a disk file. This file specifies the size and initial position of windows as well as the position of the menus. It also indicates whether the data to be displayed is to be obtained from CAMAC (the usual source) or from a data file. To customise the window sizes and their initial position to your taste, this file can be edited. The procedure is, however, error prone and will therefore not be documented here. For the benefit of the hackers, it is called WAVE\_DELTAT\_SETUP and sufficiently well commented to need no further description.

### 3.5.1 Look Top Level Menu

**<Histogram list>** displays the data for different histogram numbers from the histogram memory in CAMAC or from file and display it. A "\*" is displayed against the "current" histogram. This is the one, which was last displayed and is the one, which will be used for zooming, marker setting, background subtraction or FFT. Note that a histogram may be empty if it corresponds to a counter, which is not selected in the current "mode".

#### Asymmetry plot

**Create marker...** Markers are used to expand regions of interest in the histogram. Up to 8 markers can be created for each histogram. To create a marker, select "Create marker". A menu appears, which is described below. On the display an arrow is plotted to show the "current" position and a box giving the current bin number, corresponding time relative to bin 0 and the bin contents is displayed. The "granularity" is also shown.

**Delete...** Displays the "Delete" menu described below.

**Extras...** Displays the "Extras" menu described below.

<b>FFT...</b>	An FFT is done to the result of a "Background Subtraction" on the displayed portion of the current histogram and the result is displayed.
<b>Hardcopy...</b>	A copy of the current histogram is made in file ED0:[SLOG]HIST.PS. The file is then printed on the printer specified in ED:[COM]CUPS_HC.COM or if defined on the CUPS printer defined by the symbol CUPSDEST ::= printer_name.
<b>Move...</b>	Displays the "Move" menu described below.
<b>Background Subtraction...</b>	An exponential decay plus constant background fitted to the displayed portion of the current histogram. The difference is calculated, enhanced by an inverse exponential and displayed. The time constant of the exponential is the positive muon lifetime.
<b>Zoom...</b>	Displays the "Zoom" menu described below.

### 3.5.2 Create Marker Menu

**Abandon marker setting** does what it says.

**Increase step size** will double the granularity.

**Decrease step size** will halve the granularity.

**Bin number ...** Sets the marker to a bin value, which has to be specified..

**Time value ...** Sets the marker to a specified time value, which has to be specified..

**<<<** will move the current position to the left by a number of bins equal to the granularity. i.e. it moves a big step to the left.

**<<** will move the current position to the left by a forth of <<<.

**<** will move the current position just one bin to the left.

**>** is the opposite of <.

**>>** is the opposite of >>.

**>>>** is the opposite of <<<.

**Set marker** will set a marker at the current position.

### 3.5.3 Delete Menu

**Marker...** allows markers to be removed when no longer required. A menu is presented of the markers on the display for you to choose from. Note that markers outside the range of the display cannot be deleted. One must zoom out first.

- All markers** will cause all markers within the displayed range of the current histogram to be deleted.
- Histogram Window** This will delete one of the histogram windows. This can be useful to tidy up the display or if a continuous display is wanted (see below) and you want to display only a limited number of histograms continuously. You get a menu of histogram windows which can be deleted.
- Backgr Subtr Window** This is similar to the "Histogram Window" menu item except that the window to be deleted is that used to display the result of a background subtraction.
- FFT Window** This is similar to the "Histogram Window" menu item except that the window to be deleted is that used to display the FFT.

### 3.5.4 Extras menu

- Return to top menu** does what it says.
- Exit back to WAVE** This exits the LOOK program and returns to PV-WAVE, the underlying graphics program.  
Hitting <Ctrl-Z> in response to the WAVE prompt will return control back to DCL.
- Switch to reading data from File** It does what it says. You need to be able to see the status display window in order to operate this option. So, if it has been iconised, expand it and if it has been pushed to the background, bring it to the front again so that you can see the messages, which get displayed. You will get the menu items to help you select the data file you want, which is described below.
- Select a new Data File** You get this menu item only if data is currently being read from file (rather than from CAMAC). See the item "Switch to reading data from File" for a description of how to use it.
- Switch to reading data from CAMAC** does what it says. Whenever a histogram is displayed, data acquisition is temporarily inhibited whilst the relevant region of the histogram memory is read out.
- Continuous display** At a rate selected by the next option, all the currently displayed histogram windows will be refreshed with the latest data. The background subtraction and FFT windows will also be recomputed and refreshed if they are active.  
The continuous display is terminated by ensuring that the status display window has input focus (the banner is black or colored) and hitting any key.
- NOTE: see the above WARNING.
- Set refresh period** Sets the refresh rate for the previous option.
- Yet more options** This generates a new menu, which is described below, giving even less frequently required functions.

### 3.5.5 Move menu

<b>Marker</b>	will allow one of the currently visible markers to be moved. The marker must first be selected as under "Delete marker" and then positioned as under "Create marker".
<b>Out left edge</b>	This compresses the display by moving out the left limit of the display to the next marker.
<b>Out right edge</b>	The opposite of "Move out left edge".
<b>In left edge</b>	
<b>In right edge</b>	
<b>In both edges (zoom in)</b>	

### 3.5.6 Zoom menu

<b>Abandon zoom</b>	does what it says.
<b>Show full histogram</b>	If only a portion of a histogram is being displayed, this will zoom out so that the full histogram is displayed along with any markers, which have been set.
<b>Out</b>	This compresses the display by one level of markers.
<b>In</b>	The histogram display will be expanded so that the outer 2 markers become the left and right limits of the display.
<b>Left in</b>	The histogram display will be expanded by keeping the left limit of the display fixed and setting the right limit at the leftmost marker.
<b>Right in</b>	The opposite of "Left in".

### 3.5.7 Switch to reading data from File

**Directory is "xxx"**

**Rig is "yyy"**

**Run number is "nnn"**

**File extension is "zzz"**

**Select new directory** permits you to specify the device and directory in which the data file is located. The normal setting is ED0:[DLOG]. For runs of previous years the location ED:[DLOG.Dyyyy] may be specified.

- Select new Rig** is used to select the data files. If you want to inspect files written in TRIUMF format with the TRIUMF file naming convention, specify "TRIUMF" as the rig. Otherwise, "LTF" or "MUE1" will select data files from the current PSI rigs.
- Select run number** allows you to specify the required run number.
- Select file extension** allows you to specify the required file extension. The normal extension is DAT. For runs of previous years DAT\_yyyy has to be specified.
- Inspect previous run** decrements the run number and then operates like "Inspect this run".
- Inspect this run** will summarise information from the data file header record of the data file for the currently selected directory, rig and run number. You will then be able to select the file for histogram display.
- Inspect next run** increments the run number and then operates like "Inspect this run".
- List all files in directory** does what it says. This list can be long and therefore take a long time. It may be useful for debugging purposes.
- List all runs for this Rig** is useful for seeing which data runs are available on disk.
- Abandon selection** does what it says.

### 3.5.8 Yet more option menu

- Return to top menu** does what it says.
- Enable auto bin compression/  
Disable auto bin compression** see Note 2
- Change bin compression factor** This is only relevant if auto bin compression is disabled. It specifies the number of adjacent bins to be combined prior to display.
- Select linear y scaling/  
Select log y scaling**
- Turn on zero-suppr' n/  
Turn off zero-suppr' n**
- Rounded x scale/  
Exact x scale** Exact x scaling causes the histogram display to completely fill the axes. It has the disadvantage that the axis labels are no longer round numbers.
- Pad FFT to power of 2/  
No padding for FFT** The FFT algorithm is faster if the number of data points to be transformed is a power of 2. If padding is enabled, zero bins will be added to the data to be converted to obtain a power of 2. For large histograms, this may give a quicker FFT.
- Select linear FFT scale/  
Select log FFT scale**

**Set FFT display margins****Display box around plot/****Display no box around plot**

**Set Window Banner** will let you specify the window banner for the DECterm window, which is used for status display by Look.

**Set Muon Lifetime****Time base abscissa/****Bin number abscissa****Notes:**

1. For historical reasons, LOOK saves the TDC setup such as bin width and histogram size in its setup file prior to exiting. When it starts up again, it retrieves these values to use as a default. These may have been changed in the meantime via the "Modify/Mode" run control option, resulting in a warning message from LOOK. These messages can be ignored.
2. Displays are usually done with automatic bin compression. This causes sufficient bins to be combined together so that the number of bins to be plotted is less than but as near as possible to 1024. This saves time when doing the display. This feature can be disabled and re-enabled in the "Yet more options" menu. If disabled, the user is able to specify the actual bin compression to be used.
3. LOOK will display the histograms of all defined counters, irrespective of whether or not they are "selected". As a corollary, this means that the display of a histogram via LOOK is no guarantee that the corresponding data is actually being saved on disk. To check this, one should use the run control menu to show the current mode settings.

## 3.6 The Interpolation Programs

The interpolation programs:

```
ED0:[EXE]INTERP_MAKE.EXE and
ED0:[EXE]INTERP.EXE
```

use the cubic spline routines CSINT and CSVAL from the IMSL library. The interpolation process is a 2-step procedure. Firstly, a table of abscissae and ordinates of the curve to be interpolated must be processed by ED0:[EXE]INTERP\_MAKE.EXE to create a binary file containing a breakpoint sequence and coefficient matrix for both forward (abscissa  $\Leftrightarrow$  ordinate) and backward (ordinate  $\Leftrightarrow$  abscissa) interpolations. This binary file can then be used to perform the desired interpolations using ED0:[EXE]INTERP.EXE as described below.

Note: In this section, it is recommended that tables of values are located in directory ED0:[RCDB]. To be able to do this, you must enable the MUSR\_MANAGER right. Do this via the command:

```
$ set right/enable/attr=(res,dyn) musr_manager
```

The right can be disabled again afterwards via the command:

```
$ set right/disable musr_manager
```

By convention, the ASCII table of abscissae and ordinates of the curve to be interpolated are stored in a data file of file-type TAB, e.g. ED0:[RCDB]CGR\_123.TAB. The binary breakpoint files have file-type BINTAB, e.g. ED0:[RCDB]CGR\_123.BINTAB.

### 3.6.1 Creating the Breakpoint file

First create an ASCII file of the abscissae and ordinates of the curve to be interpolated. Blank lines and lines beginning with a "!" are ignored. Other lines may have comments appended after a "!". Each line should specify an abscissa and the corresponding ordinate. The format of the data is free so the items may be separated by any number of spaces or tabs or by a single comma. There *must* be 2 values on the line, even if the second one is zero. As an example, see the file ED0:[RCDB]THERMO.TAB.

The binary file, ED0:[RCDB]xxxxx.BINTAB can then be created via the commands:

```
$ set right/enable/attr=(res,dyn) musr_manager
$ define/user interp_in ED0:[RCDB]xxxxx.TAB
$ define/user interp_out ED0:[RCDB]xxxxx.BINTAB
$ define/user interp_test ED0:[RCDB]xxxxx.TEST
$ run ed0:[exe]interp_make
```

The file ED0:[RCDB]xxxxx.TEST is a binary file of test data generated by interpolating N uniformly spaced abscissa within the range of the .TAB data. N is set to 1000. The file contains 5 records, viz:

```
the number of data points in the .DAT file and N;
the abscissa from the input file;
the ordinates from the input file;
N evenly spaced abscissa;
the corresponding N interpolated ordinates.
```

The file can be read and displayed by PV-WAVE, for example, via the commands:

```
$ import wave
$ wave
WAVE> .run ed0:[rcdb]interp_show
Specify the file name containing test data: THERMO
WAVE> exit
```

### 3.6.2 Performing Interpolations

The interpolation program ED0:[EXE]INTERP.EXE will not only perform a forward or backward interpolation of a given curve, it will also perform a linear transformation of the interpolated value (e.g. for converting a sensor value in ohms into mV as read by a DMM). The linear transformation is applied to the result of a forward interpolation. For a backward interpolation, the value to be interpolated is first subjected to the inverse linear transformation before the backward interpolation is performed.

The result of a forward interpolation of value  $x$  is:

$$\langle \text{result} \rangle = \langle \text{const} \rangle + \langle \text{scale} \rangle * \text{INTERP}(x)$$

The result of a backward interpolation of value  $y$  is:

$$\langle \text{result} \rangle = \text{BACK\_INTERP} \{ (y - \langle \text{const} \rangle) / \langle \text{scale} \rangle \}$$

Assuming that the interpolation binary file is ED0:[RCDB]xxxxxx.BINTAB, an interpolation can be done via the commands:

```
$ interp := $ed0:[exe]interp          ! Usually in LOGIN.COM.
$ define/user interp_bin ED0:[RCDB]xxxxxx.BINTAB
$ interp <x> <direction> <const> <scale>
```

If  $\langle \text{direction} \rangle$  is zero, a forward interpolation is performed. If  $\langle \text{direction} \rangle$  is non-zero, e.g. -1, a backward interpolation is performed.

The result is returned as job logical INTERP\_DATA and as **local** symbol interp\_data specifying “ $\langle \text{input-value} \rangle \langle \text{interpolated-value} \rangle \langle \text{transformed-interpolated-value} \rangle$ ”.

Alternatively and preferably, the command procedure ED0:[COM]SET\_INTERP.COM can be used as follows to perform the interpolation:

```
$ @ed0:[com]set_interp <input> <direction> <lognam>
where
<direc>      = 0 for "forwards" and
              ⇔ 0 for "backwards";
<lognam>     specifies a logical name e.g. XXX_INTERP.
```

The result is returned as **global** symbol interp\_data.

Logical name XXX\_INTERP specifies the interpolation information required for converting a value (e.g. temperature set-point) into the physical value required for the device. It takes the form:

```
$ define/group XXX_INTERP      <file-name>, <const>, <scale>
```

where  $\langle \text{file-name} \rangle$  is the name of a binary file containing the interpolation "break-points" and  $\langle \text{const} \rangle$  and  $\langle \text{scale} \rangle$  are parameters for performing a linear transformation of the interpolated value to convert it to "useful" units. "Useful" units are deemed to be e.g. milliamps for a temperature sensor.  $\langle \text{const} \rangle$  can be used to allow for the 10 mA offset required when the CGR is the control sensor.

If in doubt, the logical name can be checked via the DCL command:

```
$ show logical XXX_interp
```

One should check that the name is defined in the group logical name table.

## 4 Internals

This section is somewhat embryonic. It is currently just a few points inspired by Ted Whidden's "MODAS at Triumph" Document. Note that the following documentation is for MODAS running on a computer directly coupled to CAMAC via a Q-Bus interface or running via *Tandem CCP* and a Starburst FEP. If the FEP is a KAV-30, the DBM protocol described in Sections 4.3 and 4.4 have been extended which is not documented here. If pTA only is used for data acquisitions Sections 4.7 and 4.8 are still valid.

### 4.1 Output Registers

The IO506 is a PSI designed CAMAC input/output register. The CAMAC output register in use at Triumph is the CES OR2088. MODAS will handle either type via simple if-then-else constructs in the code.

### 4.2 Data input to Plotdata

Plotdata reads its histogram data from disk file. The MODAS data archiver writes a data file every 5 mins during a run. If the user wants to use Plotdata to see the current state of the data in the histogram memory, he uses the deltaT menu option *Write Data to Disk*<sup>1</sup> first to get a snapshot of the data.

### 4.3 HMSR Requests

HMSR is the Histogram Memory and Scaler Readout process. It creates a multi-writer/single-reader DBM stream called *requests* for receiving requests for data from users and a single-writer/multi-reader stream called *one\_shot* for sending the requested data back again. The format of messages from a user's application to the HMSR process via the *requests* stream is:

Int*4	PID	- PID of application process.
Int*4	SUB_ID	- a message identifier chosen by the application. It is included in the replies sent by HMSR so that the application can identify the correct data.
Log*4	GET_SCALERS	- if .true., HMSR will return scaler data to the application as part of the reply.
Int*4	HM_START	- the number of the first bin wanted (the first bin is Bin 0).
Int*4	HM_NCHAN	- the number of bins wanted.

### 4.4 HMSR Data

HMSR replies to requests via the *one\_shot* DBM stream. This is a single producer/multi-consumer (i.e. "normal") stream. An application will create a pier on this stream just before sending a request via the *requests* stream and delete the pier having received the response (so that it doesn't have to keep on flushing data packets destined for other applications). The response to a request may consist of several packets (DBM calls these "events") depending on how much data was requested. The first packet's format is given by structure */reply/* in HMSR.FOR. as follows:

Int*4	PID	- the PID of the requestor.
Int*4	SUB_ID	- the SUB_ID from the request packet.
Int*4	STATUS	- the status of the request. This field could possibly indicate a CAMAC error.
int*4	NMORE	- the number of histogram data packets to follow.
Int*4	MAXLEN	- the maximum number of Int*4 bins which can fit in each histogram data packet.
Int*4	NSCAL	- the number of scalers following (zero if none requested).
Int*4	SCAL(??)	- the scaler data (if requested).

---

<sup>1</sup>This menu item is only displayed when a run is in progress.

This packet is then followed by *NMORE* histogram data packets having the structure:

Int*4	PID	- the PID of the requestor.
Int*4	SUB_ID	- the SUB_ID from the request packet.
Int*4	STATUS	- the status of the request. This field could possibly indicate a CAMAC error.
int*4	NMORE	- the number of packets still to come.
Int*4	CHAN_FST	- the histogram memory bin number of the first bin in this packet.
Int*4	NCHAN	- the number of bins of data following.
Int*4	DATA(NCHAN)	- the histogram data.

## 4.5 Computer Busy Usage by HMSR

Whilst reading out the histogram memory, HMSR sets the CAMAC inhibit line, INH, to suspend data acquisition. It removes the CAMAC inhibit once it has read the last buffer of information. HMSR should usually be able to pack all its messages into the DBM stream, even if the consumer is not reading them, and then remove the inhibit again. The effect should be a short dead-time even if, for example, the user's application is running under the Debugger.

## 4.6 SOH and HMSR

The 24 bit hardware scalers are extended in software to 32 bits. This is done by the cooperation of the 2 processes, SOH and HMSR. SOH, the Scaler Overflow Handler, sits in a polling loop with a wait time between activations of SOH\_POLL\_DLY. Usually, this logical name is not defined and so the default of 5 secs is used. If the computer busy status word, CBSW, is non-zero, SOH assumes that data acquisition is either not active or paused and so does nothing. Otherwise it checks the scaler overflow status. If set, the scalers are read and cleared and the 32 bit values, which are in a global section shared with HMSR, are updated.

Note that scalers continue to count after they have overflowed if the CAMAC INH line is not set. Hence it is important that SOH's poll rate precludes a double overflow. SOH used to be interrupt driven but this made it harder to switch between TRIUMF VDACS operation (where CAMAC interrupts go to a Starburst in the CAMAC System Crate) and MODAS operation (where CAMAC interrupts needed to go to the SCI-2280 of the VAX). In general, MODAS has avoided interrupts (at the price of aesthetics) to maintain this compatibility.

Whenever SOH or HMSR need to read a scaler, they always "read and clear" it and combine the data read with that in the global section.

The scalers (and histogram memory) are cleared by MRC behind the back of SOH and HMSR. In the future, HMSR should be extended to support a request message to do this clearing.

## 4.7 CAMAC Inhibit

The CAMAC inhibit signal, INH, is the deltaT computer busy signal. It has an associated logical name status word, *CBSW*, and an associated semaphore. If *CBSW* is non-zero, the CAMAC inhibit is set. Each process which may need to keep the CAMAC inhibit set at some time has its own bit(s) in *CBSW* which it can set or clear when it wants. When all bits are clear, data acquisition commences.

In order to change the CAMAC inhibit or its status word, one must have claimed the associated semaphore first. For "quicky" operations (e.g. scaler overflow handling), one can just:

```

claim the semaphore,
read and remember the current inhibit state,
set inhibit if necessary,
do the operation,
restore the Inhibit state and
release the semaphore.
```

For more enduring changes (e.g. *pause run*), *CBSW* is modified once the semaphore has been claimed. The

state of the CAMAC inhibit is then set according to the new value of *CBSW*. If it is non-zero, *INH* is set and *vice versa*.

The routines *CB\_LOG\_SET* and *CB\_LOG\_CLR* handle all this activity. As can be seen, the control of the CAMAC inhibit line is distributed amongst several processes. Access to it is governed by its semaphore and its setting is maintained by *CBSW*.

## 4.8 STATUS and PTASTATUS Display

The *STATUS.EXE* and *PTASTATUS.EXE* images are something of a hybrid. They can run either as a DCL-invoked foreign command or can be called from a program via the shared image mechanism. Usually the latter is how it is invoked from the deltaT run control menus in the interests of fast response. It is used as a DCL foreign command in auto run sequences to get a one-shot status display at the end of a run. The foreign command definition for *STATUS* is:

```
$ status ::= $ed0:[exe]status.exe
```

The recognised commands are then:

\$ status	generates an updating status display. The repetition rate is 10 secs. Most data is taken from group logical names. The current scaler data is obtained in the usual way from HMSR.
\$ status just_one	gives a single status display. This is used after each run of an auto run sequence for the purposes of insertion in a run log book.
\$ status scalers	Gives an updating display of the scalers. The repetition rate is 10 secs. The data is obtained in the usual way from HMSR.
\$ status scalers <xxx> <y>	does not generate a display but simply polls the current scalers waiting for scaler channel <y> to reach a count of <xxx>. This is used in auto runs to wait for the required statistics to be accumulated. To prevent accumulation of scaler counts during longer periods of no beam (e.g. during service days) you should pause the run. When the run is resumed counts accumulated in paused state will be added to the count limit. Hit <Return> to view the accumulated scaler values.

The foreign command definition for *PTASTATUS* is:

```
$ ptastatus ::= $ed0:[exe]ptastatus.exe -h host_name -e experiment_name
```

If *-h* and *-e* options are not specified the missing information will be taken from the group logical *MIDAS\_USED*

The recognised commands are then:

\$ ptastatus	generates an updating status display. The repetition rate is 10 secs. All data is read from the Midas ODB database.
\$ ptastatus just_one	gives a single status display of the pTA. This is used after each run of an auto run sequence for the purposes of insertion in a run log book.
\$ ptastatus scalers	Gives an updating display of the scalers. The repetition rate is 10 secs. The data is obtained from the Midas ODB database and the rate may fluctuate due to the inaccurate determination of the time interval. Detectors listed with * are stored in a histogram.
\$ ptastatus scalers <xxx> <y>	does not generate a display but simply polls the current scaler statistics of pTA tag <y> or if preceded by H polls the current number of events stored in histogram <y>, waiting for <y> to reach a count of <xxx>. This is used in auto runs to wait for the

required statistics to be accumulated. Hit <Return> to view the accumulated values.

This command may also help to find the maximum rate of accumulated data in the histograms.

E.g. ptastatus scalers 1000000 H1

## 5 Hardware Monitoring

### 5.1 Introduction

The term "Hardware Monitoring" is intended to mean the regular checking of the conditions of the experiment (e.g. sample temperature or magnetic field) to ensure that they are within tolerance and, optionally, to suspend data acquisition if certain bounds are exceeded. At present, two forms of hardware monitoring are possible.

- o Monitoring by purely hardware means. If some hardware is available to perform the monitoring which generates a signal when data acquisition should be suspended, this can be converted into a signal into the Inhibit input on the A2 CAMAC crate controller. Whilst the Inhibit signal is present, the TDC, scalers and histogram memory will not acquire data. A voltage comparator module exists which provides this capability.
- o Monitoring by software. This is performed by a process called MONITOR, which is started when a deltaT experiment is set up. MONITOR can periodically read out the current settings of selected devices which are to be monitored and, optionally, suspend data acquisition if the readings are outside preset bounds. MONITOR maintains a log-file of its readings so that a history plot of the monitored values can be displayed. It also maintains means and standard deviations of the measured values for saving in the data file.

This section describes the capabilities and operation of MONITOR.

## 5.2 Supported Devices

The following devices are supported by MONITOR:

- o Keithley 199 DMM with 1 input interfaced via GPIB;
- o Keithley 1992 Scanning DMM with up to 4 inputs interfaced via GPIB;
- o AVS-46 Resistance Bridge with 1 input interfaced via RS-232-C.
- o Conductus LTC-11/LTC-20/LTC-21 Temperature Controller with up to 2 inputs interfaced via GPIB.

The logical name MON\_DEVICE (see under "Setup" below) is set to a list of the device types which are to be monitored. If it is set to anything other than "NONE", MONITOR will request if it should be automatically started during the procedure of selecting the experimental rig.

At present, the program can monitor up to 8 inputs from a mixture of the above devices. The readout of a collection of Keithley 199s and 1992s is handled by a single subroutine. Up to 4 Keithleys may be read out. Lakeshore controllers are not used and therefore not supported anymore. Only one AVS-46 and up to four Conductus LTCs may be read.

During run transitions Bell Teslameter readout will be written to the monitor log file, if logical GPIB\_PRIM\_BELL7030 is defined. This logical must be defined in the instrument specific section of ED:[COM]LOGIN.COM. The LTF instrument is equipped with a Teslameter.

The GPIB connection to the computer is via a Kinetics 3388 CAMAC/GPIB interface module or LAN/GPIB server. R-S232-C connections are via one of the terminal ports on TD\_VAX.

## 5.3 Setup

Monitoring is setup by selecting the:

```

Extras ...
  Monitor ...
    Setup Monitor
  
```

run-control menu option. The current status of the setup will be displayed and the option will be given to change this. On opting to change the setup, the questions:

```

[      Do you have any Keithleys to monitor?
      and Do you have any Lakeshores to monitor?] (not used anymore)
      and Do you have an AVS-46 to monitor?
      and Do you have any Conductus to monitor?
  
```

will appear. Answering "Y" to either of the first 2 questions implies that the GPIB configuration has previously been defined. Similarly, answering "Y" to the third question implies that the RS232C configuration has previously been defined. If this is not the case, a warning message will be displayed with a hint on what to do next. In the meantime, monitoring will be disabled.

The precise details (e.g. the GPIB addresses and channel numbers) of the Conductus, Keithley and Lakeshore devices to be monitored are contained in setup files. By default, these are the files:

```

      ED0:[DAT]CONDUCTUS_SETUP.DATA
and    ED0:[DAT]KEITH_SETUP.DATA
  
```

Alternative files may be selected by defining the group logical names:

```

      CONDUCTUS_SETUP
or    KEITH_SETUP.
  
```

If a setup file does not exist, a template will be created. The user will then be given the option of editing the setup file. Sections 5.4 and 5.5 describe the setup files for Keithley and Conductus devices respectively.

The setup required by MONITOR for read-out of the AVS-46 is described in Section 5.6. This is not yet integrated into the MONITOR setup procedure.

Once the Keithley and/or the Conductus setup files have been edited, the question:

```

Specify monitoring period?
  
```

will appear. The value is in seconds and specifies the delay between monitor read-outs. Values in the range 2 to 3600 are permitted.

At this point, the new monitor status is displayed. If the user confirms that the new status is all right, the values are set and saved in the run control data base (RCDB) as part of the current *rig*.

The new MONITOR setup will take effect when MONITOR is next (re-)started, or at the start of the next run, whichever happens first. To start MONITOR, use the

```

Extras ...
  Monitor ...
    Control Monitor
  
```

run-control menu option.

## 5.4 Keithley Setup File Format

The Keithley device(s) to be read out by MONITOR are specified by a setup file located via the logical name KEITH\_SETUP. The file is read whenever MONITOR is started or restarted. MONITOR is restarted at the beginning of a run.

The file is a text file containing a sequence of records of the form

```
<verb> <param1> <param2> ...           ! comments
```

Blank records and records containing only comments are ignored. Parameters on a line should be separated by one or more tabs or spaces. Commas should not be used.

Each Keithley DMM (DigitalMultiMeter) which is to be read out is defined by a Device Definition Block (DDB). Up to 4 DDBs may be contained in the file. A DDB is started by an ADDR record and closed by an EOB record. The permitted verbs are:

**ADDR <addr>** Starts a DDB. <addr> gives the GPIB address of the Keithley.

**INPUTS <n>** <n> is the number of input channels to read from the device (max=4). Default = 1.

**CHANS <n1> <n2> <n3> <n4>** specifies the channel numbers of a 1992 scanner to be read. The number of parameters should correspond to the number of input channels specified in the INPUTS record. The channels will be read in the order specified. They do not need to be in ascending order.

**CHECK\_RANGE <n> <lo> <hi> <str>** specifies that the physical reading of input (not necessarily channel) number <n> must be checked after scaling to be within the range <lo> to <hi>. If not, <str> will be used as the basis of a DCL command which will be spawned. It is assumed that this DCL command will correct the problem. The DCL command is constructed as follows:

```
$ <str> <addr> <chans> <lo> <hi> <value>
```

where <str>, <lo> and <hi> are the values as specified on the CHECK\_RANGE record, <addr> is the GPIB address of the DMM, <chans> is the channel number of the input (useful, maybe, if a 1992 scanner is being read) and <value> is the actual value read from the channel, after possible scaling.

**EXEC <str>** The string <str> will be executed during a "restart" sequence of MONITOR. MONITOR executes a "restart" sequence when first started and at the start of every run. EXEC records are intended for initialising the Keithley, e.g. to select the range. Any "#" character in <str> will be replaced by the value of the maximum channel specified in the CHANS record (useful for setting up a 1992 scanner).

**FLIP <n> <m> <str>** indicates that input (not necessarily channel) number <n> must have its polarity flipped after every reading, usually to compensate for a zero offset. It is assumed that the polarity is changed by reversing the polarity of a Knick J152 constant current generator with GPIB address <m>. <str> specifies a group logical name whose value is the output setting of the Knick. It is usually used as part of an interpolation multiplier.

Note: A Knick J152 polarity can be flipped without having to know its actual output setting. The only change to the <str> logical name is, therefore, the addition or removal of a leading minus sign. If the logical name's value does not actually correspond to the setting of the Knick, MONITOR will not correct this problem.

- INTERP** <n> <str> This record specifies that input (not necessarily channel) <n> should be subject to a "reverse" interpolation. By default, no interpolation is done. Parameter <str> specifies a logical name which provides the interpolation information. The logical name should have 3 values, viz. the name of the interpolation breakpoint file, the constant of the linear transformation and the scaling factor of the linear transformation. Section 3.6 contains more information on the interpolation capabilities of MODAS.
- SCALE** <r1> <r2> <r3> <r4> The real numbers <r1> - <r4> specify scaling factors to be applied to the values read. The default is 1000, giving a conversion from volts to millivolts. Note that the input values are scaled before they are interpolated.
- S\_EXEC** <str> This is the same as EXEC but is only executed if the maximum channel number to be read is greater than 1. This verb is no longer recommended.
- S\_TRIG** <str> This record bears the same relation to TRIG as S\_EXEC does to EXEC. See S\_EXEC for a description.
- TRIG** <str> The string <str> will be used to trigger a read-out of the Keithley. Any "#" character in <str> will be replaced by the value of the maximum channel specified in the CHANS record (useful for triggering the 1992 scanner). The default for <str> is "T4K1X".
- TITLE** <str1> <str2> <str3> <str4> <str1>, <str2>, <str3> and <str4> specify title <str1> for the first channel, if <n2> is specified title <str2> for the second, <str3> for the third and <str4> for the fourth channel.
- EOB** Indicates the end of a DDB.
- END** Indicates the end of useful data in the setup file (so that time isn't wasted reading comments at the end of the file).

**Notes:**

- o The setup file is read whenever MONITOR executes a "restart" sequence.
- o The EXEC and S\_EXEC records are executed as they are encountered. The remaining records are saved, only the last encountered record being remembered.
- o Any "#" in a string is substituted when read, not when the EOB record is encountered. Hence, the INPUTS and CHANS records should precede any record containing a "#".
- o The end of file acts as an EOB/END record.

## 5.5 Conductus Setup File Format

The Conductus device(s) to be read out by MONITOR are specified by a setup file located via the logical name CONDUCTUS\_SETUP. The file is also similar to the KEITH\_SETUP file and is processed in a similar way. The following description will therefore just give a list of the CONDUCTUS\_SETUP verbs.

**ADDR <addr>**      <addr> gives the GPIB address of the Conductus temperature controller. This record must precede any INPUTS record. Default is 13.

**INPUTS <n1> <n2>**    <n1> and <n2> specify, which sensors are to be read.

The allowed values for <n1> and <n2> are 1 or 2:

“1” or “2” or “1 2” or “2 1” are allowed combinations for sensor 1 and/or sensor 2.

**TITLE <str1> <str2>**    <str1> and <str2> specify title <str1> for the first sensor and if <n2> is specified title <str2> for the second sensor.

**EOB**                    Indicates the end of a DDB.

**END**                    Indicates the end of useful data in the setup file (so that time isn't wasted reading comments at the end of the file).

### Notes:

- o The setup file is read whenever MONITOR executes a "restart" sequence.
- o The end of file acts as an EOB/END record.

## 5.6 AVS-46 Setup

The AVS-46 setup is not currently included in the setup procedure for MONITOR. It is usually defined in a "rig specific command procedure" associated with the *rig*. The setup, which is required, is as follows:

**AVS46\_PORT** This group logical name specifies the RS-232-C port to which the AVS-46 is connected. If the AVS-46 is connected to local port TTA2: on TD\_VAX, the definition of AVS46\_PORT would be:

```
$ define/group avs46_port local, tta2, 4800, none
```

If the AVS-46 is connected to a port on a Terminal Server and LAT service PIM3\_AV46 with password PIM3TS is associated with this port, the definition of AVS46\_PORT would be:

```
$ define/group avs46_port lat, pim3_avs46, pim3ts
```

Note that, in this case, the Terminal Server Management software must also have set the port to 4800 baud, 8 data bits, no parity.

**AVS46\_CONTROL** The AVS-46 can have a multiplexed input so that one of several inputs can be selected as the *control sensor*. The group logical name specifies the name of the sensor, which is currently selected as input to the AVS-46.

**AVS46\_<c-sensor>** This group logical name (where <*c-sensor*> is the value associated with AVS\_CONTROL) specifies the interpolation breakpoint file (see Section 3.6) for the AVS-46 control sensor. It is used for converting the value in Ohms as read from the AVS-46 into an equivalent temperature.

Currently **MEM\_MUSR: [MUSR\_TST. CHRIS. AVS46]avs\_sensor\_1tf\_setup.com** will be executed automatically if logical AVS46\_CONTROL is not defined while ED:[COM]LOGIN.COM is executed.

As an example, if the AVS-46 multiplexor input 2 is currently selected and this input is associated with sensor "RU600A2\_395" and this sensor has an interpolation breakpoint file called ED0:[RCDB]RU600A2\_395.BINTAB, suitable logical name definitions for MONITOR would be:

```
$ define/group avs46_control ru600a2_395
$ define/group avs46_ru600a2_395 ed0:[rcdb]ru600a2_395.bintab
```

Note that, in practice, the definition of *avs46\_<c-sensor>* also has other values associated with it so that, for example, the PID parameters can be set automatically whenever a control temperature is set.

## 5.7 Defining Monitoring Limits

Each time that MONITOR performs a read-out, it can compare the values obtained, after scaling and possible interpolation, against preset lower and upper limits and, if the values fall outside these limits, data acquisition will be suspended automatically. The limits are set via the commands:

```
$ define/group mon_lolim "L1 L2 L3 L4 L5 L6 L7 L8"  
$ define/group mon_hilim "U1 U2 U3 U4 U5 U6 U7 U8"
```

where L1, U1, etc. are real values. If any L is greater than or equal to the corresponding U, MONITOR will not check the value of that input channel. The upper and lower limits are defined in the RCDB and will be carried over from run to run. They are also written to the header of the data file (in PSI data files, there is only room for up to 4 values).

For example, the following definitions:

```
$ define/group mon_lolim "0 49.5"  
$ define/group mon_hilim "0 50.5"
```

if added to the values given in Section 5.12, would cause data acquisition to be disabled if the diode were to register a temperature outside the range of 49.5 to 50.5 K. The reading of the CGR would not influence data acquisition.

If the limits are changed, MONITOR must be restarted before the new values can take effect.

## 5.8 Keithley Readout

In order to read out a Keithley 199 or 1992, MONITOR first sends the GPIB command:

```
DREADING@@@X
```

so that the display "READING" appears on the DMM. This is followed by the trigger command as specified in the setup file. The recommended trigger command is:

```
B2G4I1K1T4X for a 199  
and B2G4I#K1N1#T4X for a 1992 scanner.
```

where # is the number of the highest channel to be read. It is important that the B2, G4 and K1 are present so that the read-out format is correct.

Once the Keithley has sent its data to the computer and the computer is satisfied that the data has been received correctly (MONITOR performs format checks on the data), the data will be scaled and possibly interpolated. If the data is interpolated and only a single channel is being read from the device, the Keithley is instructed to display the result of the interpolation. Otherwise, the GPIB command

```
DX
```

is sent to the Keithley to return the display to a display of the voltages.

## 5.9 Conductus LTC-20/LTC-21 Readout

The required sensors are read out by the command QSAMP?<sensor number>. Temperature of sensor 1 is read out by the command QSAMP?1; and temperature of Sensor 2 by the command QSAMP?2; . The readout answer string is then checked if temperature units are K. If -1.0 is returned as temperature value the sensor table may not be assigned to the sensor, the sensor may not be connected to the temperature controller or the sensor reading may be out of range.

## 5.10 MONITOR Example

Suppose that there are 2 temperature sensors in the cryostat, a carbon glass resistor and a GaAlAs diode. The CGR is assumed to be supplied with a current from a Knick J152 (note that the current through a CGR must be adjusted as a function of temperature so that the voltage across the resistor does not exceed 3 mV). The diode is assumed to be supplied with a fixed current of 10 microamps. The voltages from the sensors are monitored by 2 Keithley 199 DMMs. The current through the CGR must be inverted after each reading to allow for a zero offset. Assume the GPIB addresses of the Keithleys are 27 and 28 and that of the Knick is 9. Assume further that the program which sets the current in the Knick also sets the logical name CGR\_MICROAMPS to indicate the setting.

The following setup file would define this configuration:

```

===== Start of Device Definition Block for CGR DMM =====
ADDR 27                ! GPIB address of the device.
INPUTS 1                ! Number of Inputs to Read (max = 4).
CHANS 1                ! Channels to read.
SCALE 1000             ! Scaling Factors (get milliVolts).
INTERP 1 KEITH_CGR     ! Interpolate the value to get degK.
FLIP 1 9 CGR_MICROAMPS ! Flip CGR current after each reading.
TRIG B2G4I1K2T4X      ! Read-out trigger.

EXEC B2X                ! Buffer. MUST BE B=2 !!!
EXEC F0X                ! Function
EXEC G4X                ! Data format.
EXEC I1X                ! Data store size.
EXEC K2X                ! K2 = (EOI and bus hold-off on X). MUST BE K=2 !!!
EXEC M0X                ! Mask for Serial Poll
EXEC P2X                ! Filter
EXEC Q0X                ! Interval.
EXEC R0X                ! Range
EXEC S1X                ! Scale
EXEC Y0X                ! Y0 = <Cr> <Lf> terminator.
EXEC T4X                ! Trigger Mode.

EOB                    !===== End of Device Definition Block =====

===== Start of Device Definition Block for Diode DMM =====
ADDR 28                ! GPIB address of the device
INPUTS 1                ! Number of Inputs to Read (max = 4)
CHANS 1                ! Channels to read
SCALE 1000             ! Scaling Factors (get milliVolts)
INTERP 1 KEITH_DIODE   ! Interpolate the value to get degK.
TRIG B2G4I1K2T4X      ! Read-out trigger.

EXEC B2X                ! Buffer. MUST BE B=2 !!!
EXEC ...                ! etc. as for the previous DDB

EOB                    !===== End of Device Definition Block ===== END

```

In addition, the logical names KEITH\_CGR and KEITH\_DIODE must be defined to obtain a successful interpolation of the voltages. Suitable definitions might be:

```

$ define/group keith_cgr          ed0:[rcdb]cgr_8941.bintab, -
                                0.0, -
                                0.001*cgr_microamps

```

and

```
$ define/group keith_diode ed0:[rcdb]diode_8421.bintab, -  
    0.0, -  
    1000.0
```

Note the commas and minus signs in these commands. The verification of the values of the linear transformation parameters in these commands is left as an exercise for the reader!

## 5.11 In Case of Problems

Problems can arise if a device is turned off while MONITOR is accessing it, a power failure occurred leaving the hardware in an undefined state or when the set-up to MONITOR a new GPIB device was not done properly.

Quit all programs that show device settings or are used to set field, temperature or flow.

Because MONITOR runs as a background job, it can be difficult to see the cause of any problems which may be present. The background job has a log file called

```
ED0:[SLOG]MONITOR.LOG.
```

To view the last 200 lines of the file enter the command

```
$ TYPE ED:[SLOG]MONITOR.LOG /TAIL=200.
```

If the log file documents that the MONITOR has problems, use the

```
Extras ...
Monitor ...
Control Monitor ...
Exit
```

menu option to get MONITOR to exit. If MONITOR refuses to exit immediately in response to this selection<sup>1</sup>, it may be necessary to wait (maybe a very long time) or to use brute force and issue the command:

```
$ STOP monitor or
$ SHOW SYSTEM and $ STOP /ID=PID_of_monitor
```

The following steps require that MONITOR now stopped. Verify using \$ SHOW SYSTEM.

The menu option

```
Extras ...
Monitor ...
Show Monitor Status
```

may be used to inspect the monitoring setup. Taking the information from the log file, try to find the device that is not working properly. E.g.: Conductus sensor(s) not connected.

If there is an indication that a single device is causing all the problems, remove the device from being monitored using "Setup Monitor" and start MONITOR. However, if all parameter seem to be all right and removing devices from the set-up does not help either, there may be a GPIB problem. Make sure MONITOR is turned off.

Check if a process is holding the GPIB and/or CAMAC semaphore

```
Define the foreign command $ semutil := $ED:[EXE]sem_utility
$ semutil show /semaphore=GPIB and $ semutil show /semaphore=CAMAC
```

If the semaphore is blocked (e.g. Waiting for user input) try to locate the process and terminate the process or kill (STOP /ID=pid) the process to free the semaphore.

If the problem is not due to a blocked semaphore, follow the GPIB cables of the attached devices and verify if they are connected and powered on.

Read the name of the LAN/GPIB gateway and check the name with the name of the GPIB\_SERVERx group logical.

E.g.: \$ SHOW LOG /GROUP GPIB\_SERVER\*

Devices in the address range 032 are accessed using ONC RPC communication to GPIB\_SERVER, addresses between 1000 and 1032 to GPIB\_SERVER1 and between 2000 and 2032 to GPIB\_SERVER2).

Telnet to the respective GPIB server and enter status<Return> (on Agilent E5810) or status <Ctrl>J (on

---

<sup>1</sup>MONITOR sometimes gets stuck in an error-retry loop if there are serious GPIB problems.

Agilent E2050; command will not be echoed) to see if there are any pending RPC connections. Type quit<Return> or quit<Ctrl> J to leave the Telnet session. If a GPIB server may not be reached it may be either blocked or the last telnet connection was not freed. To unblock it turn the power of the GPIB server off and then on again.

Execute “Extras .../GPIB & RS232C Setup .../Show GPIB/RS232 Setup” to get a list of currently configured GPIB devices. Try to set (“Modify .../Set Device .../”) or show each GPIB device separately.

To be able to readout a Keithley type 199 define the symbol

```
$ gpibs ::= $ED:[EXE]gpibs gpib_server_name
```

where gpib\_server\_name is the IP name of the GPIB server. To read the measured voltage enter

```
gpibs “receive” xx                (xx is the GPIB address of the Keithley).
```

A response of the form:

```
Received "NDCV-000.1595E-3,C0="      should be obtained for a Keithley type 199.
```

If you are not able to access GPIB devices turn off all GPIB devices and the LAN/GPIB server. Turn the LAN/GPIB server on first and try to telnet to the server to get the status as described above. If there are no problems, subsequently turn other devices on and use show or set to check if they are accessible. Try to verify the GPIB address in use when each device is started up to make sure that there is no address conflict and the device addresses are set as they are expected to be. Check Table 2.2 for default assignments of GPIB addresses.

If all devices may be reached try to set-up and start MONITOR again.

If all the above tests run successfully and there are still problems, try a *recover* as described in Section 2.2.

## 6 Conductus LTC-20 Temperature Controller

For Conductus set up and setting the cryostat set points the following menu option is used:

```
Modify ...
  Set Device ...
    Conductus
```

If the Conductus menu option is missing in the Set Device ... menu, the GPIB address of Conductus and/or the Kinetics 3388 CAMAC to GPIB interface have to be defined first in:

```
Extras ...
  GPIB & RS232C Setup
```

Use the GPIB Setup to define the GPIB address of the second Conductus temperature controller. Menu options for the 2<sup>nd</sup> Conductus will then be available.

How to set up Conductus LTC-11/LTC-20/LTC-21 and download sensor calibration tables and PID tables is described in <http://lmu.web.psi.ch/intranet/ltc.html>. Additional help is available during execution of Conductus set up.

The following files should be available on TD\_VAX for Conductus to be operable:

ED:[DAT.LTC]LTC_<serial_number>.TAB	Conductus memory lookup table containing information about downloaded sensor calibration tables and PID tables.
ED:[DAT.LTC]SAMPLE_HOLDERS.TAB	Sample holder table containing sensor names, default sensor channel assignment and PID table name for each sample holder.
ED:[DAT.LTC]<sensor_name>.LTC	Sensor calibration table(s) downloaded during device initialization.

NOTE: Sensor type definitions for LTC-11 and LTC-2x differ. Verify calibration table before downloading.

ED:[DAT.LTC]<pid_table>.PID	PID table(s) downloaded during device initialization.
-----------------------------	---

If there are problems after Conductus has been set up or during a run

```
Show ...
  Conductus
```

may give diagnostic information. Sensor calibration table names are read back and compared with the information in the memory lookup table.

If there are severe problems communicating with Conductus try to stop the Monitor process first. This is described in section 5.13.

## **7 AVS-46 Resistance Bridge & TS520 TC**

## 8 EL734 Step Motor Controller

To be able to operate the EL734 step motor controller the group logical EL734 and the global symbol el734 have to be defined.

The group logical EL734 defines the instrument (or rig) name and the motor names in use. The global symbol el734 specifies the EL734 executable and RS232 terminal server parameters to be used.

Group logical and global symbol for each instrument are defined in the instrument specific part of ED:[COM]LOGIN.COM. This is usually done by the instrument scientist.

```
$ SHOW LOG /GROUP EL734   and   $ SHOW SYMBOL /GLOBAL el734
```

```
PW251> show log /group EL734
      "EL734" = "DOLLY ANGLE X" (LNMSGROUP_000301)
PW251> show symbol /global el734
      EL734 == "$EDO: [EXE]EL734.EXE psts218e 3002 10000 -v"
```

MODAS run control may then be used to set angle rotation and X position. The DCL command procedure ED:[COM]EL734\_THINGS.COM is then executed.

When powered on EL734 is in local mode. The yellow square buttons are then lit. To set a motor manually, select the green (round) button of the motor first and then use the yellow button to position. Fast positioning is selected when the green button of the selected motor and the yellow button are pressed at the same time.

**NOTE:** After remotely positioning a motor, EL734 will be put into local mode again. If this is not the case, press the green square button on top for more than 3 seconds to allow manual positioning again. (This may happen if remote setting was repetitively interrupted by local user interaction!)

**NOTE:** The angle position will only be updated in the run title if the angle is remotely set using the MODAS run control menu.

## 9 Facility Instruments at PSI

Up to date information about location, set-up and usage of facility instruments is available on the  $\mu$ SR specific PSI web pages (see <http://lmu.web.psi.ch/facilities/facility.html>).

## 10 Ortec pTA 9308

The hardware setup for pTA 9308 is shown in Figure <pTA Setup Fig>.

To get additional help about the pTA operating system and the front end application start-up execute command

```
@ED:[COM]MIDASRC HELP PTASTARTUP
```

on the DCL level.

To view the last message written by the front end execute command

```
midasset -1 -c show mode
```

on the DCL level or view the application log of the Windows operating system selecting Programs->Administrative Tools->Event View in the Start menu on the pTA front end console.

To enable (y) or disable (n) debug information that may be viewed on the pTA front end console execute the command

```
midasset -1 -c set mode dbg y
```

on the DCL level when the front end is not running or is in state stopped.

The front end then has to be shut down and (automatically) restarted executing command odbedit on the DCL level.

```
odbedit
```

```
[PCxxxx:td_musr:S]/> scl
```

enter scl to get the front end name

```
Name          Host
```

the front end name should be similar

```
pta_test      PCxxxx
```

to the front end's module name

```
ODBEdit      PC3229
```

```
[PCxxxx:td_musr:S]/> shutdown pta_test  shutdown front end
```

```
[PCxxxx:td_musr:S]/> exit          exit odbedit
```

On the DCL level execute the command

```
@ED:[COM]MIDASRC TEST
```

to verify if MODAS and Midas are in equivalent run states and have the same run number.

## 11 Glossary

- \$** Default prompt of the DCL command interpreter when waiting for DCL command input.
- Active Window** See *Input Focus*.
- BE** data acquisition Back-End.
- CCP** *CCP* is the Tandem CAMAC Command Processor, which is used to perform CAMAC cycles in configurations with either a Starburst or KAV-30 front-end processor. A CCP server runs in the FEP and accepts DECnet connections from a CCP client running on the TD\_VAX. The CCP client is invoked via the DCL command, *CCP*. Use the DCL command *help ccp* for more information.
- DBM** *DBM* is the Data Buffer Manager. It is a *Tandem* sub-system which allows inter-process communication via global sections. *Streams* are conceptually set up in these global sections via data structures and processes can create *piers* on the banks of the stream for the insertion of data into the stream or the retrieval of data from the stream. See CSN-87-028 for full documentation.
- DCL** Digital Command Language the (default) command line interpreter of the OpenVMS operating system.
- FE** data acquisition Front-End.
- FEP** *FEP* refers to the Front-End Processor. This is either a KAV-30 rtVAX-300 CPU housed in a VME crate and connected to CAMAC via a CBD-8210 VME/CAMAC interface or an ACC-2180 Starburst CPU housed in CAMAC.
- Halt Button** The *Halt Button* on a VAXstation 3100 is located next to the Ethernet connector on the back of the CPU box. On a VAXstation 4000, it is located behind a flap on the front of the CPU box. On a microVAX III, it is located on the front of the CPU box.
- Pressing the *Halt Button* causes the console monitor prompt, ">>>>" to appear on the computer console.
- HMSR** *HMSR* is the Histogram Memory and Scaler Readout process. It runs on TD\_VAX as a background ("network") process. It sets up a DBM stream for accepting requests for a read-out of HM and/or scaler data from other programs on TD\_VAX. The requests may also be control actions such as setting up the resolution of the TDC. It then accesses CAMAC to acquire the requested data or effect the requested action and responds to the requestor via another DBM stream.
- Input Focus** On a window display, only one of the windows can be selected at any given time to receive characters typed on the keyboard. This window is said to have *input focus* or to be *active*. It is highlighted in some way. Input focus is usually given to a window by clicking on it with <MB1>. It is also possible using the Window Manager menus to set up DECwindows so that input focus follows the mouse pointer.
- KAV-30** A *KAV-30* is a VME module manufactured by C.E.S. of Geneva and marketed by DEC. It incorporates an rtVAX-300 CPU and sufficient memory to run the VAXeln operating system. There is a connector on the front for connection to an Ethernet transceiver. There are also 2 RS-232-C connectors. The upper one is the main console which can be useful for debugging if a dumb terminal is connected to it.
- The usual MODAS configuration for the KAV-30 is for it to be linked to the  $\mu$ SR CAMAC crate via a CBE-8210 VME CAMAC branch driver. The KAV-30 boots

its operating system from TD\_VAX over the Ethernet.

To reboot the KAV-30, the spring loaded *Reset/Halt* switch on the front of the module should be pressed to the *Reset* position and released. The KAV-30 will execute its diagnostic test procedures (the 2 hex display digits count down) and then boot. When booting is complete, the 2 hex digits should be zero. The total boot procedure should take about 20 seconds.

**<MBn>**

<MB1>, <MB2> and <MB3> are the left, center and right mouse buttons, respectively (unless a left-handed pointer has been selected in which case <MB1> and <MB3> are swapped)..

**Midas**

refers to the Maximum Integrated Data Acquisition System (see <http://midas.psi.ch>). Midas was designed as data acquisition system for medium and small particle physics experiments at PSI and was chosen to replace the TANDEM data acquisition system used at PSI. Midas is based on TCP/IP socket communication and runs under LINUX, UNIX, VMS, VxWorks, Windows NT and MS-DOS operating systems. To prepare the migration of MODAS from TANDEM on VMS/VAX to another operating system new data acquisition equipment used by MODAS (e.g. pTA FE) was developed based on Midas components and Midas run control.

**MODAS**

refers to the Muon Online Data Acquisition System being the main subject of this documentation. MODAS is based on TANDEM components.

**Power Switch**

The *Power Switch* on a VAXstation 3100 is located on the back of the CPU box. On a VAXstation 4000, it is located on the front of the CPU box.

**pTA 9308**

refers to the Midas Front End. A Windows NT system running the Midas front end application that is designed to operate and read out the Perkin Elmer pTA 9308 picosecond Time Analyzer.

**PWxxx>**

Default prompt of the DCL command interpreter when logged in the MUSR\_TST account on VAXstation PSWxxx (e.g. PSW254) waiting for a DCL command (e.g. td or deltat) to be entered.

**SOH**

*SOH* refers to the Scaler Overflow Handler process. On TD\_VAX configurations with either a direct Q-Bus connection to CAMAC or a Starburst FEP, SOH runs as a background ("network") process on TD\_VAX. On KAV-30 configurations, SOH runs as a child process of the HMSRV server. In either case, it operates by polling the scalers to check for an overflow. If a scaler overflows, the overflow is noted and reset. For reasons of compatibility with TRIUMF, SOH has avoided using interrupts, although this would have been more elegant.

**Starburst**

A *Starburst* is the common name for the ACC-2180 module manufactured by C.E.S. of Geneva. It is a single width CAMAC module containing a PDP-11/70 compatible processor and sufficient memory to run the RSX-11S operating system. The connectors on the front are for connection to LSI-11 Q-Bus compatible devices. The usual configuration at PSI is for the Starburst to be linked to a DEQNA (Ethernet interface) and an SCI-2280 (CAMAC System Crate Interface). The Starburst boots its operating system from TD\_VAX over the Ethernet. To reboot the Starburst, a 50Ω terminator is inserted into the RSTR socket and removed. Note that DECnet must be up and running on TD\_VAX before the Starburst can be booted.

**TANDEM**

*TANDEM* refers to the data acquisition "toolbox" developed by the Computer Group at PSI and aimed primarily at the particle physics experiments around the cyclotron. Certain items of TANDEM are used by MODAS; in particular, DBM (Data Buffer Manager), CMR (Central Message Reporter), CCP (CAMAC Command Processor) and the experimental user group directory structure conventions. There is extensive documentation available from the Computer Group on these items.

---

<b>TD_VAX</b>	<i>TD_VAX</i> refers to the main deltaT (Time-Differential) OpenVMS/VAX system for a given experimental area.
<b>TD_WS</b>	<i>TD_WS</i> referred to a slave VAXstation, which was used in certain areas.
<b>TDC</b>	Time to Digital Converter

## **12 Appendix**

## 13 Index

### A

accessing run files, 21  
auto run control, 39  
automatic run sequences, 24  
AVS-46 resistance bridge, 69  
AVS-46 set-up, 60

### B

begin run, 37  
boot procedure, 9

### C

CAMAC configuration, 38  
computers, 5, 8  
Conductus LTC readout, 63  
Conductus LTC set-up, 59  
Conductus LTC temperature controller, 68  
continue data taking, 37

### D

Data File Location on Linux BE, 36  
Data File Location on OpenVMS, 36  
data files, 36  
define or check rig parameters, 34  
DELTAT, 37  
deltat menu, 37  
disk quotas, 21

### E

EL734 Step Motor Controller, 70  
executing an auto run sequence, 25  
extras menu, 39

### F

facility instruments at PSI, 71  
FEP, 6

### G

GPIB, 30  
GPIB set-up, 39

### H

hard-copy, 18  
histogram display, 42

### I

interpolation programs, 48  
interrupting a run during an auto run sequence,  
26  
interrupting an auto run sequence, 25

### K

Keithley readout, 62  
Keithley set-up, 57  
kill run, 39  
killing an auto run sequence, 27

### L

LAN/GPIB gateway, 6  
Linux Midas backend, 6  
log auto run sequences, 17  
logging in, 16

### M

Midas, 6  
Midas set-up, 40  
MODAS, 6  
modify menu, 39  
modify mode settings, 34  
modifying an auto run sequence, 26  
monitor, 40  
monitor problems, 66  
monitor setup, 56  
monitoring limits, 61

### N

network, 5

### O

Ortec pTA 9308, 72

### P

pause data taking, 37  
preparing an auto run sequence, 24  
printing, 29  
pTA FE, 6  
pTA mode menu, 40  
pTA status display, 52

### R

real run, 35  
recover procedure, 11  
remotely starting an auto run sequence, 25  
RS232, 32  
RS232 terminal server, 6  
RS232C set-up, 39  
run control, 37  
run numbers, 35

**S**

secondary beamline client, 6  
secondary beamline server, 6  
session manager setup, 28  
show menu, 38  
shutdown, 15  
start MODAS system, 9  
status display, 52  
stop waiting for instrument during an auto run  
sequence, 26

**T**

TANDEM, 7  
TD\_VAX, 6  
TDC mode menu, 40  
terminate run, 37  
test run, 35  
trouble shooting, 13  
TS520 temperature controller, 69

**W**

write data to disk, 37