Operation Manual for μSR at TRIUMF

Syd R. Kreitzman — *TRIUMF*
Donald J. Arseneau — *μSR Facility*
Bassam Hitti — *μSR Facility*
Ted Whidden — *TRIUMF*
Jess H. Brewer — *UBC*

TRIUMF
4004 Wesbrook Mall,
Vancouver, B.C.
Canada V6T 2A3

Revised: Oct 2, 1998

Contents

Introduction to μSR Data Acquisition, Data Display, and Data Analysis Software at TRIUMF

Since the early 90's the μSR data acquisition at TRIUMF has run on VAX workstations under the VMS operating system. At the same time TD-muSR, which was originally written by David Maden at PSI, was imported and adapted to the TRIUMF environment, and the I-muSR program was ported from the PDP-11 version. TD-muSR and I-muSR control their respective primary data acquisition functions and are independent from programs that control and monitor different instruments, and from the display and analysis programs. These Data acquisition programs are described in Chapters 1 and 2.

At TRIUMF the interface to control and monitor instruments (temperature controllers, magnet power supplies, NMR or Hall probes) is handled by a separate program called CAMP, described in Chapter 3. A document that more thoroughly describes CAMP, how it can be configured and used, will soon be available.

The TRIUMF μSR plotting routines are written in PLOTDATA and can be used to display the saved data from the current or (and) previous runs. These graphic programs are also utilized to determine time zero, first and last good bins, to look at the background, perform simple fitting and Fourier transforms. The *μSR Utilities in PLOTDATA* document written by Jess H. Brewer provide a complete description of the PLOTDATA routines available to display μSR data. It is abbreviated here in Chapter 4. The LCRplot program, for display of I-μSR runs is described in Chapter 5.

TD-muSR, I-muSR, CAMP, LCRplot, and PLOTDATA are normally run on the data acquisition computer. Data analysis programs are available on the VAX cluster under the names MSRFIT, or various $*$FCN. MSRFIT, in particular, is specifically designed for analyzing a wide range of μSR signals to several types of filtered μSR data, and, being a complex system, is difficult to learn. Therefore there is a new graphical front-end for MSRFIT called Xmusrfit. Printed documentation to these powerful programs will soon be available, and they already provide on-line help.

1 Collecting Time-Differential Data: TD-muSR

'`TD-muSR`' is the time differential μSR run control program used on all the μSR data acquisition computers at TRIUMF. TD-muSR is a menu driven program that allows the user to configure the experiment, start a run, monitor its progress and finally stop and save the data.

To start TD-muSR from VMS type TD then press Return.

This document introduces new users to the various functions of the TD-muSR program. We start by explaining what is meant by the words "`REAL`", "`TEST`", "`Rig`", and "`Mode`" which appear in the title bar, and continue with descriptions of many TD-muSR commands. A few commands are only available when a run is in progress (or not in progress); some can be executed at any time; and some will lead to further menus (these commands are followed by "..." in the menu list). This document is organized in the same order used to structure the menus. Some of the menus, commands, and displays have changed for VME-based acquisition and the B980 TDC, but they are subject to further revision as the acquistion system stabilizes, and this documentation does not yet reflect reality in all detail.

1.1 Title Bar

Figure 1.1 shows the TD-muSR main menu with no run active; the settings used in the previous run are displayed. "`REAL`" in the top right corner of the title bar refers to a real run, as opposed to a "`TEST`" run. After the experiment is

```
TD_muSR  Run Control                REAL
 Rig is "OMNI_FBLR", Mode is "FB1.25"
Run not active. Last run was 5904
```

```
Begin run



Show . . .
Modify . . .
Extras . . .
Comment Log . . .
Help
Quit
```

Figure 1.1: TD-muSR main menu display when run not active.

over, real data are copied and archived on the main VAX cluster. Test runs are normally used during setup, while tuning the electronics and beam, to make sure that all the elements of the experiment are functioning correctly. They are not archived. An active test run can be changed to real, and a real run to test, by using the '`Modify... Mode... Acq`' command described below.

The second line of the title bar shows the 'Rig' and 'Mode' used during the last run. The Rig contains all the CAMAC interface configuration parameters and information concerning all the counters that are set up. The Rig needs to be modified if, for example, the counters and the data acquisition electronics have changed. By modifying the Mode one can select the counters of interest for a particular measurement and define time-zero, first, and last good bins, the number of bins per histogram, and the TDC resolution (the time per bin). In the example shown above, the Rig '`OMNI_FBLR`' refers to the μSR magnet OMNI with forward, backward, left and right counters, while the Mode '`FB1.25`' implies that only the forward and backward counters are in use, and a TDC resolution of 1.25 nsec per bin (code 4) is selected. Both Rig and Mode names are arbitrary, so if you create a new mode, for example, you should choose a descriptive name in the same style as the existing names. Both the Rig and Mode need to be set up from the '`Modify`' menu before the first run is started.

```
TD_muSR  Run Control              REAL
Rig is "OMNI_FBLR", Mode is "FB1.25"
Run 5905                 <<< ACTIVE >>>
```

```
Terminate run
Pause data taking
Write data to disk
Show . . .
Modify . . .
Extras . . .
Comment Log . . .
Help
Quit
```

Figure 1.2: TD-muSR main menu display when run is active. In comparison with Fig. 1.1, note that '`Begin Run`' has changed to '`Terminate run`', and there are two additional entries to '`Pause`' and '`Write`'.

1.2 Main Menu

To select an item from any of the menus, either move the cursor line to that item and press Return, or type the first character of that item (*e.g.*, "`B`" to Begin a run). At the start of a new run the run number is automatically incremented by 1, and the user will be asked to enter information about the sample, temperature, magnetic field, run title, the initials of the people on shift, and the experiment number. When the run is started, the run statistics are reset and the main menu changes to that shown in Fig. 1.2. Note that the '`Begin run`' entry has changed to '`Terminate run`', which will end the current run. There are also two newly available commands: '`Pause data taking`' and '`Write data to disk`'. Selecting '`Pause data taking`' will suspend the data acquisition until the '`Continue`' command is issued. The data are automatically saved to disk every 5 min, but it is possible to save at any time during the run with the '`Write data to disk`' command. Only data that have been written to disk can be plotted or analyzed.

1.3 Show

The '`Show... Run Status`' command (Fig. 1.3) gives up-to-date information on the current run, including titles, scaler values (both current values and as last saved), and the total number of events saved in each histogram. If a run is active

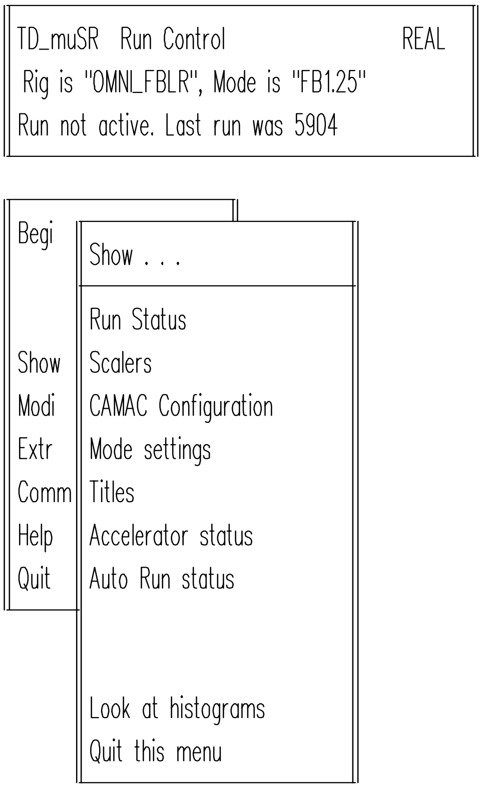


Figure 1.3: Show, run not active

then the same command should be used to monitor the counts in each of the selected counters and the total number of events collected in all histograms, the run statistics are updated as the data is written to disk. When there is no run active, the final statistics of the last run are displayed.

From the show menu one can also check the counts in the individual scalers, CAMAC configuration (a subset of the Rig), Mode settings (Fig. 1.8), and the run title. Selecting '`Accelerator status`' allows the user to display messages from the control room, the beam schedule, or cyclotron information. The same information can be obtained from VMS by the '`cystat`' command. The '`Auto Run Status`' command can be used to check on the progress of an automatic run sequence. As will be explained later at TRIUMF automatic running of μSR experiments has not found much use.

The menu item '`Look at histograms`' invokes PLOTDATA to display run data.

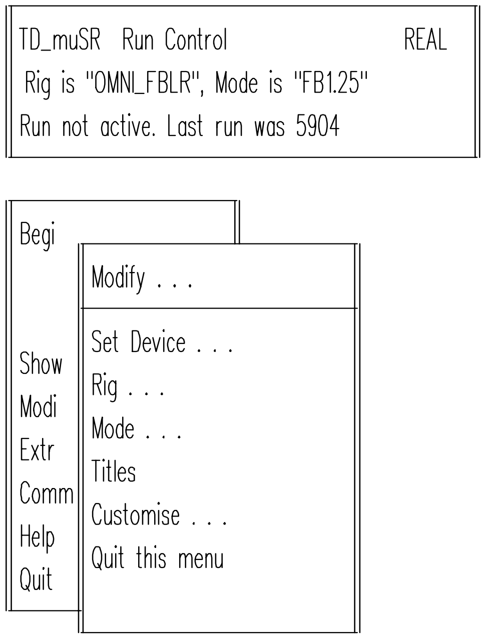


Figure 1.4: The '`Modify...`' menu, with run not active

1.4 Modify

The commands in the '`Modify...`' menu fall into three groups: commands that can be executed when data is not being collected, those that are active only when a run is in progress, and those that can be used at any time.

1.4.1 Modify commands when run not active

The modify commands that can only be executed when a run is not active are '`Modify... Set Device...`', '`Modify... Rig...`' and '`Modify... Mode...`' (Fig. 1.4).

The '`Modify... Set Device...`' command is intended to control and monitor instruments. At TRIUMF this function is handled by the CAMP program and, therefore, the above command is not used.

As explained earlier the 'Rig' specifies the CAMAC interface parameters. Normally, the '`Modify... Rig...`' command is used only after a major change in the apparatus at the start of an experiment; it results in the '`Rig Setup...`' menu shown in Fig. 1.5. '`Select Rig`' gives a list of the saved Rigs and the option to load one of the files. One can then use the '`Define or Check Rig Parameters`' command to check, and if necessary, modify, the setting by accepting the defaults or answering a series of questions concerning the position of the various electronic modules in the CAMAC crate. An example list of settings and the minimal dialog ("No No No") resulting from the '`Define or Check Rig Parameters`' command

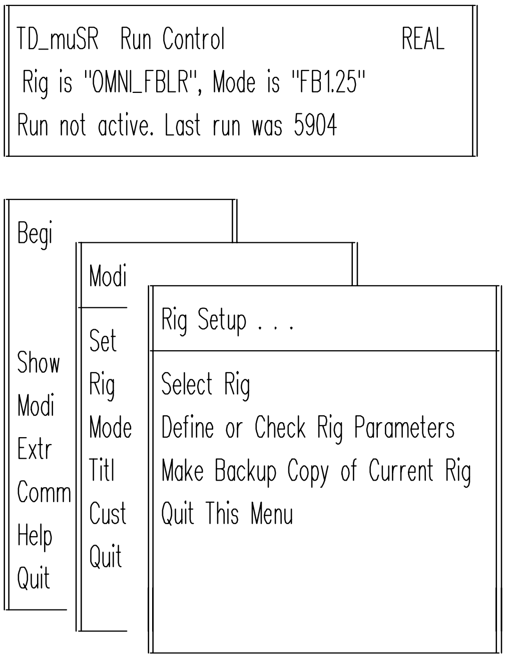


Figure 1.5: The '`Modify... Rig...`' menu. This is only available when there is no run active.

is shown in Fig. 1.6. An important rig parameter is the type of histogramming memory: LRS, CES, or VME. The first two options go with the LRS 4204 TDC residing in CAMAC; the third indicates a BNC B980 TDC residing in a VME crate, in which case the 'histogramming memory' is the main memory in the MVME CPU.

The '`Modify... Mode...`' menu is shown in Fig. 1.7. Most of the commands in this menu are related to loading and modifying a preset mode. '`Modify Mode Settings`' is used to select the relevant counters and appropriate TDC resolution. The TDC resolution codes vary from 1 (0.15625 nsec per bin, nominally) to 15 (2560 nsec per bin) according to the formula

$$bin\_size = 0.078125\,\mathrm{ns} \times 2^{code} \ ,$$

e.g., code 4 gives bins 1.25 nsec wide and code 5 gives 2.5 nsec bins. The dialog resulting from the '`Modify Mode Settings`' command is shown in Fig. 1.8.

1.4.2 Modify commands when run active

When a run is active the commands to zero or kill the run replace the '`Set Device...`' and '`Rig...`' commands in the modify menu (see Fig. 1.9). The '`Zero Run`' command erases the data saved in the histogramming memory and resets the run statistics before the run is started again. '`Kill Run`' ends the run

```
CAMAC Interface is an SCI-2280 in Station 21.
CAMAC Branch = 0
CAMAC Crate  = 0

TDC 1 is in Stations 8 and 9
Histogram Memory Type is CES in Station 7 6 0 0.
Singles Scaler   Type is K3615 in Station 17.
Scaler Labels are:
  "Us"  "Ug"  "Fg"  "Bg"  "Lg"  "Rg"
Output Register is an OR2088 in Station 2.
There is no Averaging Starburst.

 If you change any of these CAMAC parameters, the experiment will have
 to be shut down. This can take quite a while and is a relatively rare
 operation.

Do you want to change the CAMAC parameters ["N"]?

Current Counter Definitions

              NIM/ECL     OR2088
 Counter       Input      Output
 =======      =======     ======
 Bg              1           1
 Fg              2           2
 Lg              3           3
 Rg              4           4

Do you want to change the Counter parameters ["N"]?

The Current Mode is "BFLR1.25".

Do you want to select a new Mode ["N"]?
```

Figure 1.6: Selecting either of the menu items ‘`Modify... Rig... Define or Check Rig Parameters`’ or ‘`Show... CAMAC Configuration`’ gives a list like this. Pressing Return at each prompt (as in this sample) leaves all parameters unchanged.

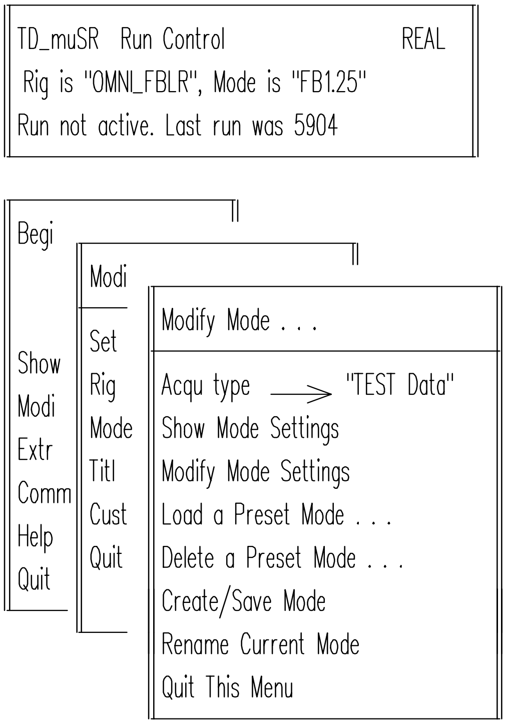


Figure 1.7: Modify mode

```
Current Mode is "OMNI_FB1.25"

                     NIM/ECL   OR2088
  Counter  Selected   Input    Output   T-zero  First   Last
  =======  ========  =======   ======   ======  =====   ====

  F           Yes       1         1        252    258   8190
  B           Yes       2         2        252    258   8190
  L           No        3         3
  R           No        4         4

Number of selected Counters   = 2
Maximum histogram in Hist Mem = 2
Bins per Histogram            = 8192
TDC Resolution                = 4 (1.25 nsec)
TDC Time Overflow             = 63 (10160 nsec)
Histograms cover time range from 0 to 10.24 usec

Hit <Return> to acknowledge:
```

Figure 1.8: A typical list of information provided by the menu items '`Show... Mode settings`', by '`Modify... Mode... Show Mode Settings`', and by '`Modify... Mode... Modify Mode Settings`'. In the case of '`Modify`', the final line is instead "`Is this OK ['N']?`"; if you then press Return, you are prompted to change each parameter.

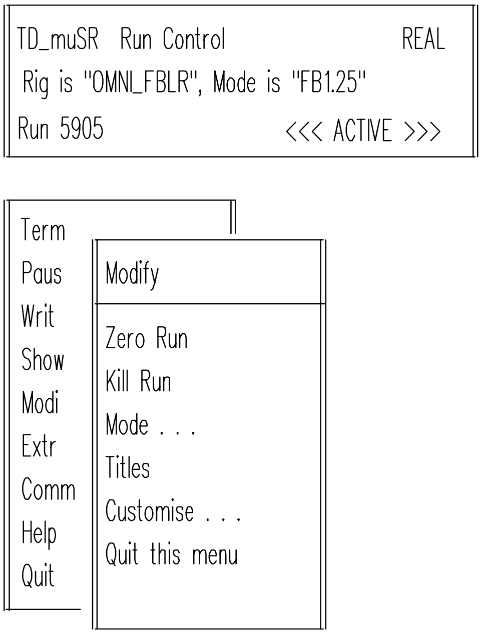


Figure 1.9: Modify, run is active

without saving the data, and deletes any previously-written data files for this run.

With the run active, the '`Modify Mode Settings`' command can be used only to enter and save time-zero and first and last good bins for each of the selected counters. (There are μSR PLOTDATA routines to determine the correct values.)

1.4.3 Modify commands which are always available

The top item in the '`Modify Mode`' menu allows the user to change the data acquisition type between "`REAL Data`" and "`TEST Data`". Test data are collected during the setup process and are given run numbers above 30000. At the end of the experiment the real data will be copied to a disk on the main data analysis cluster while the test data will just be deleted from the data acquisition machines.

The commands available under the '`Modify... Customise...`' menu are shown in Fig. 1.10. These commands are available so that the users can alter some of the features of TD-muSR to suit their preference.

1.5 Extras

The '`Auto Run Control...`' command of the '`Extras`' menu shown in Fig. 1.11 can be used to prepare and begin an auto run sequence or to check on the progress of a sequence that is running. To make the best use of the available beam we

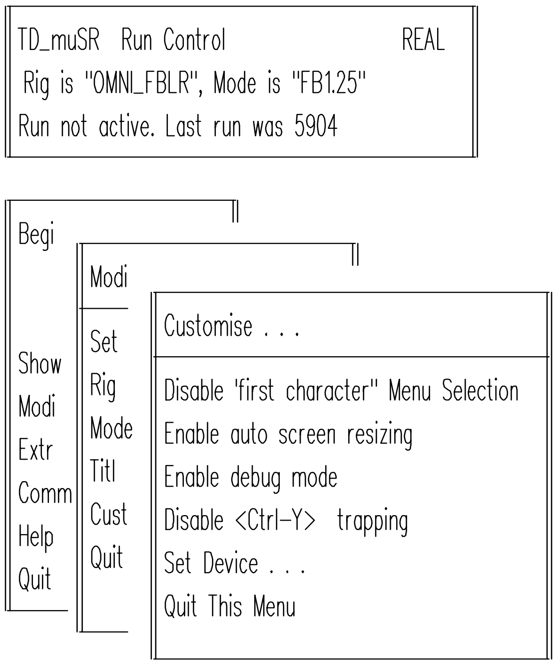


Figure 1.10: Modify customise

do not encourage users to run in this mode; an experiment should be attended at all times. Otherwise, any malfunction will result in many hours of lost beam. Additionally, in most situations the experiment can make more efficient use of the beam when data are analyzed on-line before the decision on how to proceed with the measurement is made.

As stated earlier μSR uses the program 'CAMP' to handle all the communication to the control and monitor instruments. Therefore, there is no need to use the TD-muSR '`GPIB & RS232C Setup...`' or '`KS4580 Setup...`' commands. CAMP can be called from the '`Extras`' menu shown above or from VMS by the '`camp`' command.

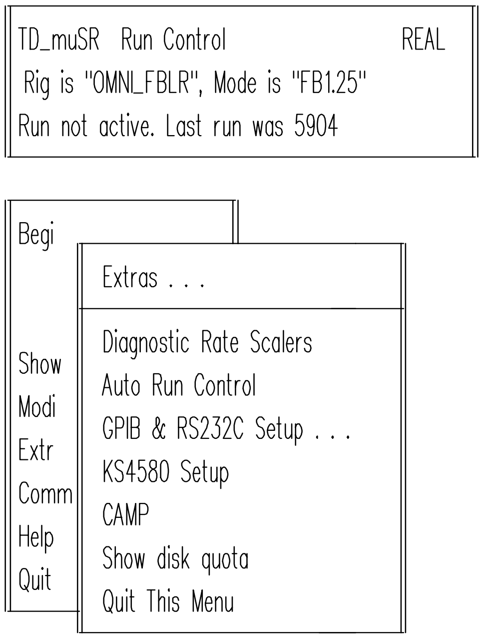


Figure 1.11: Extras

2 Time-Integral μSR Data Acquisition

"I-muSR" is the time integral μSR run control program used on all the μSR data acquisition computers at TRIUMF. It was written by Syd Kreitzman to replace the more primitive PDP-11 data acquisition software when the μSR data acquisition at TRIUMF was moved to VAX workstations under the VMS operating system. I-muSR is a menu driven program that allows the user to configure the experiment, start a run, monitor its progress and finally stop and save the data.

The operation of I-muSR is similar to that of the TD-muSR program used at TRIUMF to control the time differential data acquisition. This section provides an introduction to the operation of the I-muSR program, and is organized in the same order used to structure the menus.

To start I-muSR from VMS type IMUSR and press Return.

To select an item from any of the menus either move the cursor line to that item and press '`Return`', or type the first character of that item. When there is an ambiguity—more than one command starting with the same letter—typing a letter will only move the cursor; you must press '`Return`' to execute the command

2.1 Define Hardware

After starting I-muSR the menu shown in Fig. 2.1 is displayed. The main function of this menu is to define the hardware used in the experiment with

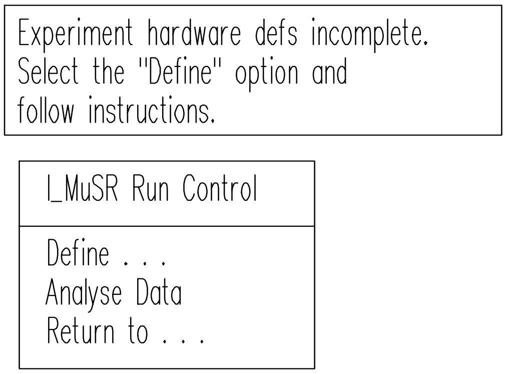


Figure 2.1: The initial I-muSR menu.

the ‘`Define...`’ command. The two other commands ‘`Return to...`’ and ‘`Analyse Data`’ appear in this and some of the other menus. ‘`Return to...`’ will always produce a menu with the options to return to DCL or the top menu. The command ‘`Analyse Data`’ is intended to be used in the future for on-line data analysis; currently this command is not used. ‘`Define...`’ is used to define the experiment hardware, this command when issued results in the menu shown if Fig. 2.2. The ‘`I-MuSR Set Up`’ command is used to set up the device that

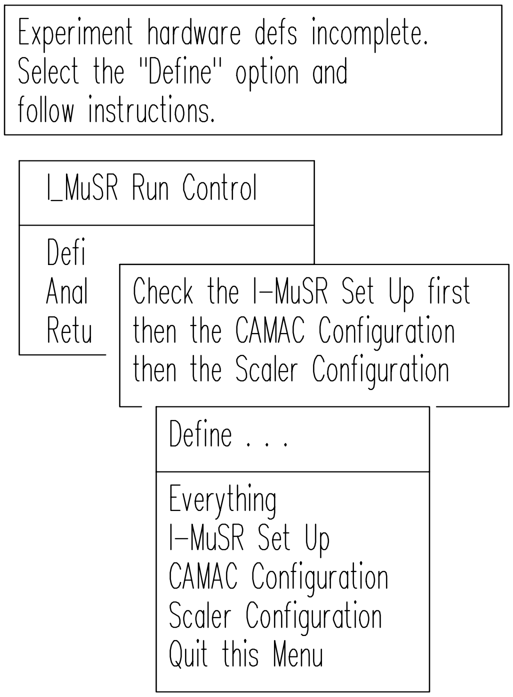


Figure 2.2: The ‘`Define...`’ menu before I-muSR acquisition has been configured.

```
RC_CHOICE IS: DEFINE SET_UP
A set up is defined by a list of devices.
The first is the primary device which is "Scanned".
The remaining secondary devices may be "Set".

Old set up was HELIOS
Possible devices are: HELIOS   DAC   RF   TEMP

Define your set up:
Specify RS232 port for HELIOS power supply: ["TXA1:"]
Remember, for helios supercon, the max current is 70000 mA.
For other magnets, you may get 120000 mA out.
Specify the maximum Helios current allowed (mA): [70000]
Use the internal calibration for the Helios magnet? [N]
Do you wish to calibrate the Helios PS? yourself? [N]
Experimental set-up has been defined should be checking out Camac.
Hopefully Camac would check out, experimental state being put to READY.
```

Figure 2.3: Sample '`I-MuSR Set Up`' dialogue.

will be scanned. For LCR experiments this will be the power supply used to scan the magnetic field (magnet current for Helios or DAC for Omni). In an RF experiment one can either scan the field or the frequency synthesizer (RF). An example of the '`I-MuSR Set Up`' dialogue is shown in Fig. 2.3.

The CAMAC is configured by using the '`Modify-Rig Setup`' command of the TD-muSR program. The '`CAMAC Configuration`' command of the I-muSR program merely displays the following reminder:

```
RC_CHOICE IS: DEFINE CAMAC
You must define the TD-MuSR set before defining IMuSR.
```

The '`Scaler Configuration`' command allows the user to define the scaler input for the '`FORWPSCALER`' (forward plus scaler), '`FORWMSCALER`' (forward minus scaler), '`BACKPSCALER`' (backward plus scaler), '`BACKMSCALER`' (backward minus scaler), '`TOTALSCALER`' (The sum of all scalers; sometimes connected to a muon counter instead) and '`CLOCKSCALER`' (clock used for normalization). These scalers are used to determine the integral asymmetry (the quantity of interest in an integral experiment) as a function of field or frequency. The integral asymmetry is calculated from the following equation:

$$A \equiv \frac{F^+ - F^-}{F^+ + F^-} - \frac{B^+ - B^-}{B^+ + B^-} \tag{2.1}$$

The plus and minus refer to the RF-on and RF-off in an RF experiment and to the polarity of the flip coil in an LCR experiment.

```
RC_CHOICE IS: DEFINE SCALE-CONFIG
Specify the FORWPSCALER input: [1]
Specify the FORWMSCALER input: [2]
Specify the BACKPSCALER input: [3]
Specify the BACKPSCALER input: [4]
Specify the TOTALSCALER input: [5]
Specify the CLOCKSCALER input: [6]
```

2.2 Acquisition configuration

The very first main menu is displayed in Fig. 2.1, where the only functioning option is to define the hardware configuration. After that is done, the standard main menu will be offered; it is shown in Fig. 2.4.

Before starting data collection the acquisition parameters should be configured using the '`Acq Config`' command from the main menu, Fig. 2.4. As seen in Fig. 2.5, the user can choose the number of toggles and sweeps, the preset counts and tolerance.

Toggles are the number of times for each field (frequency) point that the experiment is in the plus and minus state. If the toggle number is 1 then, for each set point, data will be measured once in the plus state and once in the minus state. If a value of 0 is entered, no toggles will be performed.

```
RC_CHOICE IS :ACQCONFIG TOGGLES
Specify number of toggles (0 = SMOOTH sweep mode): [0]
Specify the toggle settle time in millisec: [100]
```

The number of presets determine the counts in each state per toggle. So if the number of presets is 6 and the preset scaler is set to 1 million then 6 million

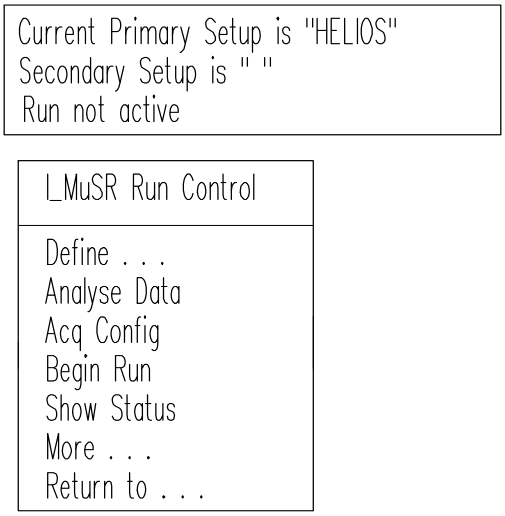


Figure 2.4: The standard main menu of I-muSR, with run not active.

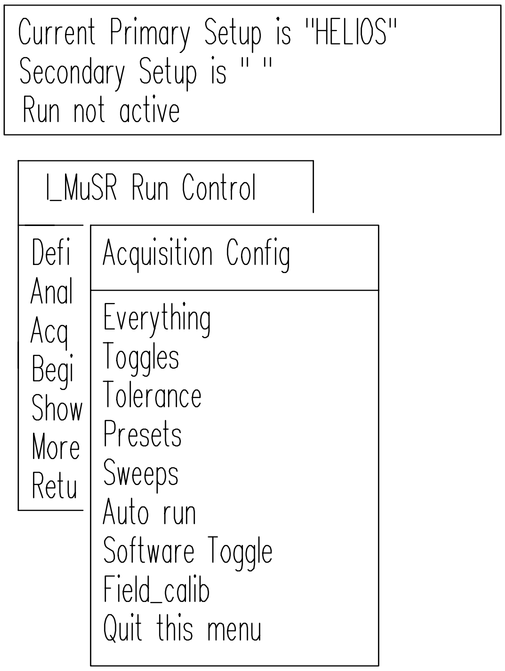


Figure 2.5: The menus after selecting '`Acq Config`' from the main menu.

events will be counted in each of the plus and minus states before the next toggle is started or the magnet ramped to the next field point.

```
RC_CHOICE IS :ACQCONFIG PRESETS
Specify number of presets: [6]
```

The tolerance command allows the user to choose the acceptable percentage variation in data rate before the experiment is put on hold. After the beam current is back in tolerance the experiment is continued automatically. The "data rate" is the ratio of the Totals-scaler to the Clock-scaler, and it can vary considerably as the magnetic field is swept due to solenoid focussing. If you use sweeps over a large field range, then you should '`Reset Normalization`' occasionally, or set a large Tolerance, say 20%.

```
RC_CHOICE IS :ACQCONFIG TOLERANCE
Input the rate tolerance. [5]
Input acquisition delay (sec) after coming back into tolerance. [20]
```

Finally, the number of sweeps is the total number of scans to perform before the run is ended. If the sweep number is set to zero then the run will continue until it is stopped by the operator.

```
RC_CHOICE IS :ACQCONFIG SWEEPS
Specify number of sweeps (0 = continuous): [0]
```

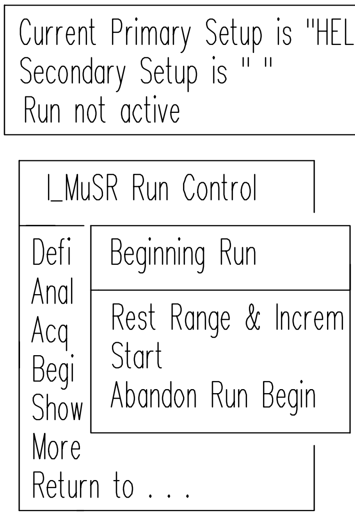


Figure 2.6: The menu of options after selecting '`Begin Run`'.

2.3 Begin Run

A new run is started with the '`Begin Run`' command of the main menu. A new range and step size can be chosen or the run can be started with the old settings, Fig. 2.6. The '`Reset Range & Increm`'(ent) command results in the dialogue shown in Fig. 2.7. Note that some of the phrasing is misleading: In that sample, as is often the case, the so-called "DAC in bits" values are really "current in milliamps".

```
RC_CHOICE IS :RUN RESET
Specify the STARTVAL of the DAC in bits. [20505]
Specify the STOPVAL of the DAC in bits. [20845]
Specify the INCREMENT of the DAC in bits. [10]
Specify the field step settle time in millisec: [100]
Is the scan range OK?  [Y]
Do you want a REAL or TEST run? ["TEST"]
Getting the run number.
Sample: ["165torr C6D6+N2=16atm"]
The scan range is 20505-20845:10
Description: ["80C"]
comment1: ["no comment"]
Is the title & comment & scan range information OK? [Y]
Updating RCDB
Beginning acquisition of run 30053 165torr C6D6+N2=16atm 20505-20845:10
```

Figure 2.7: Sample dialog after selecting '`Reset Range & Increm`' when beginning a new run.

```
Current Primary Setup is "HELIOS"
Secondary Setup is " "
Run 5000 is paused
```

```
I_MuSR Run Control
------------------
Analyze Data
End Run
Continue Run
Change Parameters
Show Status
More . . .
Return to . . .
```

Figure 2.8: The I-muSR main menu right after a run has started and is still in the paused state. Alternatively, this menu is also shown when a run has been deliberately paused in the middle.

At the start of a new run the 'Begin Run' command puts the run in a PAUSED state. The operator should enter the 'Continue Run' command to start the data collection.

After receiving the '`Continue Run`' command, the status line in Fig. 2.8 changes to, *e.g.*, "Run 5000 in Progress", and the '`Continue Run`' menu item (Fig. 2.8) is replaced by the '`Pause Run`' command.

2.4 End Run

The normal way to end a run in progress is with the '`End Run`' command followed by the '`YES (finishes point)`' option; Fig. 2.9. The '`Quickstop`' command gives the user the option to end the run destructively without saving the data.

2.5 More: Helios Special Set Up

The '`More... Helios I_MuSR Specials...`' command results in the menu shown in Fig. 2.10. From this menu one can remotely set the ramp rate and maximum current of the Helios power supply.

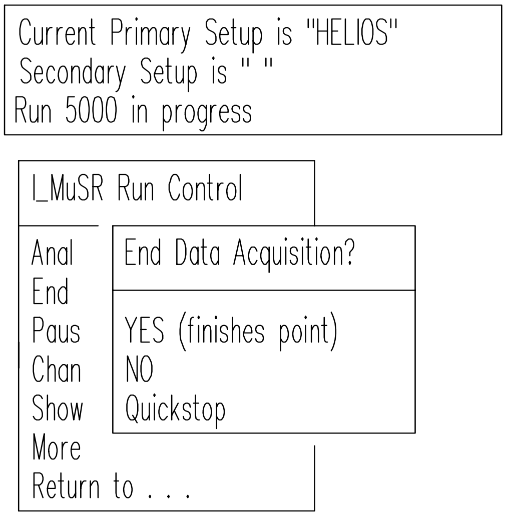


Figure 2.9: I-muSR menu when ending a run.

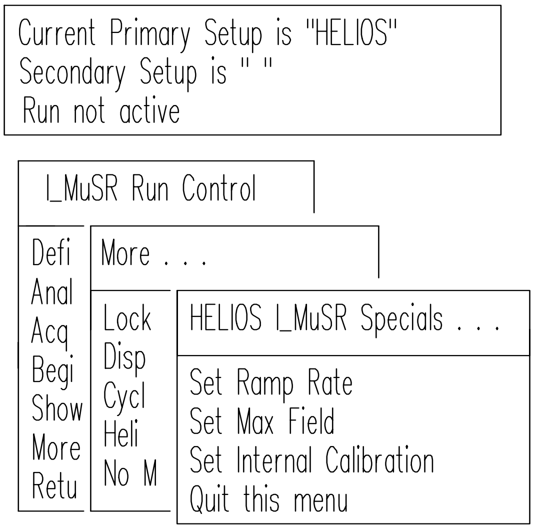


Figure 2.10: The '`More... Helios I_MuSR Specials...`' sub-sub-menu for configuring the Helios power supply.

3 The CAMP Slow Control System

3.1 Introduction

The CAMP (Control And Monitoring of Peripherals) Slow Control System provides experimental users and experimental software with a central, consistent method of accessing all peripheral instrumentation. Users configure the CAMP database for their particular experiment, and then use the CAMP User Interface to control and monitor the instrumentation. Typical applications include: magnet power supply control and temperature controller monitoring and control. Likewise, CAMP is also used by data acquisition software that requires the control and monitoring of instrumentation. Applications include: experimental magnet control and NMR field control.

Each experimental instrument, even those providing the same functionality, has a different set of commands for setting and retrieving parameters. Instruments often have subtleties and idiosyncracies that can cause problems. CAMP handles and hides these differences with one consistent interface. CAMP becomes particularly invaluable in cases where an instrument has more functionality via remote control than front panel control, or when instrumentation is located a great distance from the experiment operator. Furthermore, there are devices in use that have only analog control, and must be set remotely using a DAC (Digital to Analog Converter).

3.2 Capabilities

CAMP may be used with intelligent instrumentation that have standard RS-232-C or GPIB interfaces, or with analog devices (using a DAC). The following is a list of currently supported instruments:

- BIRD Model 4421 RF Power Meter (RS-232-C)
- Cryogenic Consultants PS120C-H Series-C Power Supply (RS-232-C)
- Group3 DTM-141 Digital Teslameter (RS-232-C)
- Hewlett Packard HP-59303a DAC (GPIB)
- Joerger D/A-16 DAC (CAMAC)
- LakeShore 330 Autotuning Temperature Controller (RS-232-C, GPIB)
- Lakeshore 450 Digital Gaussmeter (RS-232-C, GPIB)
- LakeShore 622 Magnet Power Supply (RS-232-C)
- Metrolab PT3020 NMR Teslameter (RS-232-C, GPIB)
- Oxford Instruments PS120-10 Magnet Power Supply (RS-232-C)
- Solartron 7061 Systems Voltmeter (GPIB)

CAMP has been designed to allow for easy integration of new instruments. Requests for implementation of a driver for a new instrument can be fulfilled with short notice and are welcomed. Please submit requests to the μSR Software Manager: Suzannah Daviel, local 7306, e-mail `suz@triumf.ca`. Please indicate the date required, and where she can obttain the appropriate (programming) documentation for the instrument.

CAMP is currently used on the VxWorks and VMS Operating Systems at TRIUMF, but was written with portability in mind, and could easily be ported to other operating systems.

3.3 Using CAMP

3.3.1 Startup

A textual user interface has been provided to access the CAMP database online. Experimental users begin the interface by typing the command `camp`:

```
$ camp
```

Note that starting the CAMP User Interface does not start the CAMP System. The CAMP database is maintained by a process that is always present in the background. The user interface is used to retrieve information from and make requests to this background process. Likewise, exiting from the user interface does not stop CAMP. Multiple copies of the user interface may be run at the same time, allowing multiple users access to CAMP from anywhere on site or from home. This is particularly important when expert users must intervene in an emergency. Easy access has not yet been a problem for security, but there is a facility to lock instruments for exclusive control by one user.

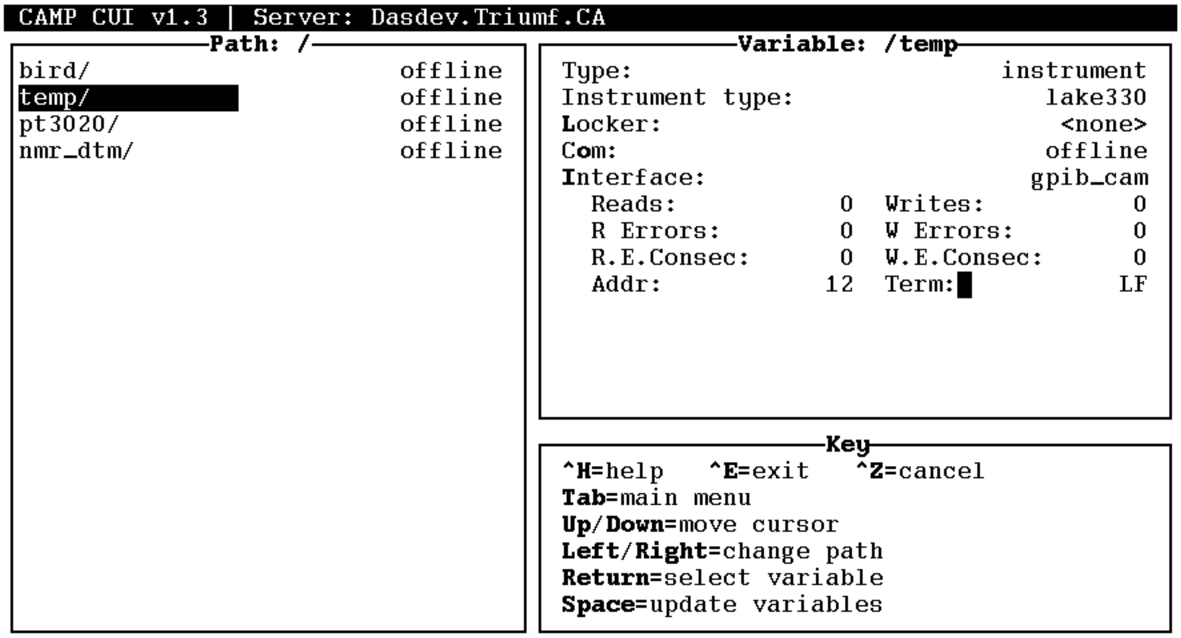


Figure 3.1: Startup screen.

3.3.2 Viewing the Instruments and their Variables

When the CAMP User Interface is started, a screen similar to Figure 3.1 will be seen. There is a list of variable names on the left, a window of variable information in the upper right, and a key window in the lower right.

Instruments and their associated variables are organized like a directory tree. The instrument names are listed at the *top* of the tree. This list is what is seen when the interface is started. Associated variables are located *below* the instrument variable. Use the up and down arrow keys to position the cursor over the instrument of interest, and then use the right arrow key to move to a list of associated variables. The list of instruments is then replaced by a list of associated variables (Fig. 3.2). In fact, any variable name ending in the / character indicates that there are more variables below (like a directory of variables). Use the left arrow key to move back up a level in the tree.

The variable information window in the upper right always shows detailed information about the variable that is highlighted by the cursor. The line at the top of this window shows the full pathname of the variable.

3.3.3 Instruments

The list of instruments currently accessible from CAMP is shown at startup (Fig. 3.1). Users may add, delete and modify the entries in this list online.

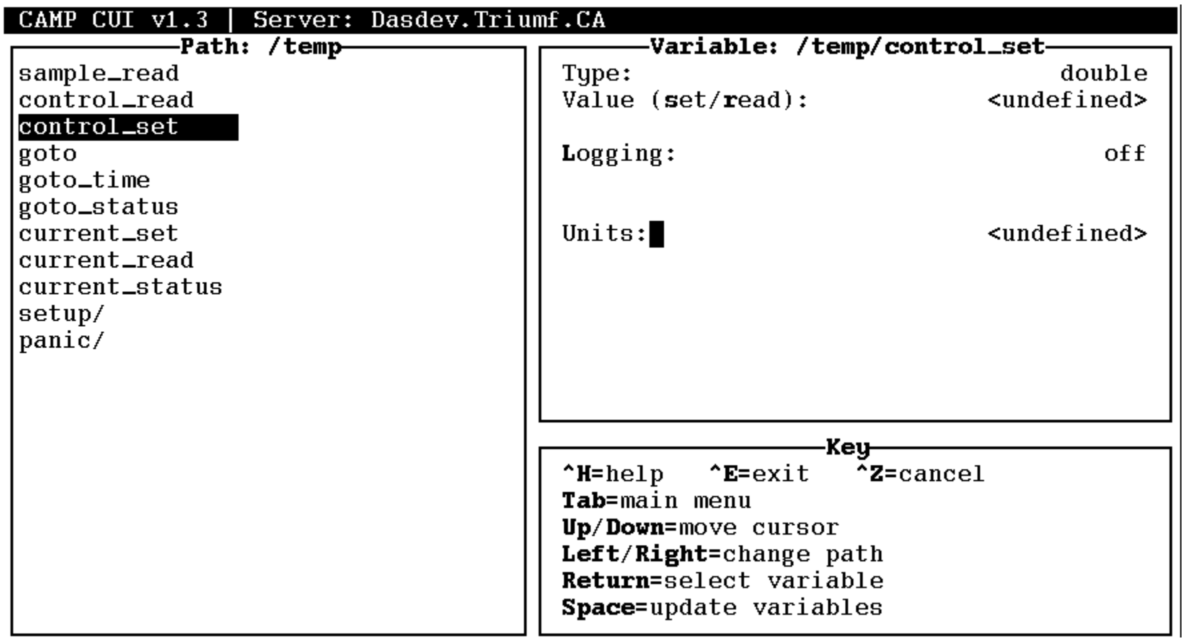


Figure 3.2: Display after moving down a level in the tree.

3.3.3a Adding

To add an instrument, press the `Tab` key to bring up the main menu. Select `Configure` from the Main Menu, and then select `Add instrument` from the Configure menu (use up and down arrow keys to move the cursor). You will then be prompted for the instrument type (select from a list), unique instrument keyword (choose something descriptive), interface type and interface parameters. The default interface type will be highlighted in the list of possibilities; this will be the normal choice. For the interface parameters, you should normally select the defaults for all but the GPIB address, RS-232-C portname and CAMAC slot number. If you do not know this parameter for your instrument, or whether the instrument has even been connected, find a μSR contact.

NOTE: Press `<Ctrl>-Z` to cancel the currently displayed menu or input window at any time.

After this information has been input, CAMP will add the instrument to the database, and verify that communication is possible (thus setting the instrument *online*.

Instrument interface information may be reconfigured at any time. First, make sure that the interface is *offline*. This status is shown both in the variable list window and the variable information window. Then, select the instrument with the cursor, press `Return`, and select the `Interface` selection from the menu shown in the center of the screen. You will then be prompted for the interface type and parameters.

3.3.3b Online/Offline

Setting an instrument *online* verifies that communication is possible and allows communication. Setting an instrument *offline* disallows communication. To toggle this parameter, select the instrument with the cursor and press `Return`. A menu of actions will be seen in the middle of the screen. Select the `Online` or `Offline` entry from the menu.

3.3.3c Locking

To lock an instrument for exclusive control, select the instrument and choose the `Lock` selection from the action menu. Lock status is shown in the variables information window in the upper right.

3.3.3d Deleting

To delete an instrument, use the `Tab` key to bring up the main menu. Choose `Configure` and `Delete instrument` and then select the instrument to delete from the list that will be shown in the middle of the screen. You cannot delete an instrument that is locked by another user.

3.3.4 Variables

Instruments have many associated variables. The most frequently used variables are located in the level below the instrument. Other, less often used, variables might be located at lower levels. Variable types include: *double*, *integer*, *selection*, *string* and *structure*. The *selection* type has a value that is a selection of a list of strings. The *string* type has a character string value. The *structure* type is a container for more variables *below* it in the tree (like a directory).

3.3.4a Setting

To set a variable, select it, and then choose `Set` from the action menu. You are then prompted for input appropriate to the variable type.

3.3.4b Updating

To update the value of a variable, choose `Read` from the action menu. The instrument will be interrogated for a current reading.

3.3.4c Polling

Variables may be set to be updated at a regular interval. Choose `Polling ON` from the action window and then input the interval in seconds. Choose `Polling OFF` to toggle.

3.3.4d Logging

Variables may be saved in the data acquisition data file. Choose `Logging ON` from the action window, and then choose the type of data acquisition experiment that you wish to log the variable (i.e., `TD-MuSR` or `TI-MuSR`). For TD-μSR experiments, the variable and its statistics are saved when the file is written. For I-μSR experiments, the variable's value is logged for each data point taken. It therefore does not make sense to toggle logging on and off during an I-μSR experiment.

3.3.4e Alarms

Variables may be set to trigger an action when the value goes out of tolerance. Choose `Alarm ON`, and then select the action. Possible actions include: halting TD-μSR (`halt_TD-MuSR`) and halting I-μSR (`halt_TI-MuSR`). For *double* variables, you are also prompted for the percent or plus/minus tolerance value.

4 On-line Display of TD-μSR Data

This document provides a brief introduction to the PLOTDATA macros written by Jess Brewer to manipulate and display time-differential (TD)-μSR data. Only the frequently used display commands (no fitting) and a few useful PLOTDATA commands are described. For a more complete description of these programs refer to the document *μSR Utilities in PLOTDATA* written by Jess Brewer, the LaTeX input file for this document is on USR1:[MSR.PLOTDATA]PD_MUSR.TEX.

4.1 Introduction

To utilize the μSR PLOTDATA macros it is useful but not necessary to become familiar with the PLOTDATA program. For information about PLOTDATA, users are referred to the TRIUMF manuals *"PLOTDATA Users' Guide"* and *"PLOTDATA Command Reference Manual"* written by Joe Chuma.

During the experiment, PLOTDATA is normally run on the data acquisition computer. The μSR macros are used to determine time zero, first good bin, last good bin, the background bin range, and to display the data in time or frequency space. While several types of fitting are built into these display macros, it is recommended to perform all the fitting on the cluster where more sophisticated programs are available. The μSR display routines can also be used on the cluster. To run PLOTDATA and the μSR macros on the cluster refer to the document *"μSR Utilities in PLOTDATA"*.

On the data acquisition machines, PLOTDATA is setup to start with an initialization file (PLOTDATA.INIT), defining all the μSR commands described below.

To start the PLOTDATA program just enter 'PLOTDATA'.

Help facilities for both PLOTDATA and the μSR macros are available on-line, they can be invoked by the HELP and HLP commands, respectively.

4.2 Data Files

The current TD-μSR data files are stored on the data acquisition computer in the directory ED:[DLOG]. The data from the current run (run in progress) is saved to disk automatically every few minutes. The user can also at any time use the 'Write data to disk' command of the TD-muSR data acquisition program to save the data, as described in Chapter 1. Only saved data can be read, processed and displayed by PLOTDATA and the μSR macros.

To select a run enter:

RUN ⟨*Run_Number*⟩

from within PLOTDATA.
For example: `RUN 5001`
Information that describe the data are also stored in the file, and after the file is loaded this information becomes available in the following variables.

- `IRUN` is the Run number, a scalar.
- `TITLE` is the Run Title, a text string.
- `NHISTS` is the number of Histograms, a scalar.
- `NBINS` is the number of Bins per histogram, a scalar.
- `NS_BIN` is the time (ns) per Histogram Bin, a scalar.
- `IT0` is the file-specified bin number where $t = 0$ occurs, a vector of dimension `NHISTS`.
- `IT1` is the file-specified bin number where "good" data starts, a vector of dimension `NHISTS`.
- `IT2` is the file-specified bin number where "good" data ends, a vector of dimension `NHISTS`.
- `IH` contains the actual Histogram data, a matrix of dimensions `NHISTS` × `NBINS`. The $\mathtt{i}^{\text{th}}$ Bin of the $\mathtt{j}^{\text{th}}$ Histogram is in the element `IH[i,j]`.

Use `HOW RUN` to list the current values of these variables or use the `=` command in the form "`=`⟨*symbol_name*⟩" to inquire for the value of one symbol; *e.g.*, "`=NHISTS`".

4.3 Display Control Variables

Some of the μSR macros process the data before it is displayed. For example the command `ASY` displays $A(t) \equiv [F(t)-B(t)]\ /\ [F(t)+B(t)]$ for each *pair* of Histograms $F(t)$ and $B(t)$. The vector `JHST` tells PLOTDATA the order of the Histograms to use when performing such an operation. If `JHST={1,4,2,3}`, then the `ASY` command will display two asymmetry spectrums $A(t) = [1-4]\ /\ [1+4]$ and $A(t) = [2-3]\ /\ [2+3]$. Here is a list of PLOTDATA variables that control the data display.

Scalars, [*e.g.*, `NPAC=10`]

- `FRQ1` Lower limit on FFT frequency range (MHz).
- `FRQ2` Upper limit.
- `NHMAX` Maximum number of Histograms to process (default `NHMAX=6`).
- `NPAC` Packing factor (default `NPAC=0`, meaning PLOTDATA will choose the packing to give fewer than 256 packed bins).

Vectors, [*e.g.*, `JHST={1,4,2,3}`]

- `IBG1` First bin of $t < 0$ background.
- `IBG2` Ditto—last bin
- `JHST` Correct sequence of Histograms. The order in `JHST` also defines the pairs of histograms used for calculating asymmetries.
- `NT0` Any number other than 0 will override the $t = 0$ bin specification (`IT0`) in the data file (default `NT0={0,0,0,0,0,0}`).
- `NT1` ditto—first good bin.
- `NT2` ditto—last good bin.

The value of any scalar or vector can be obtained by the PLOTDATA command "`=`" with the syntax "`=`⟨*symbol_name*⟩"; *e.g.*, `=NPAC` or `=JHST`.

Generally, about 200 ns delay is added to all the 'stop' pulses from the positron telescopes in the TD-μSR electronics in order to allow accumulation of counts from *before* the muon entered the target—*i.e.*, before time zero. This provides an estimate of the random background in each histogram. The PLOTDATA variables `IBG1` and `IBG2` define the first and last bin of the region used to determine this uncorrelated (flat) background. You can use the `BKGD` command (macro) to automatically assign values to `IBG1` and `IBG2`.

Normally the time zero, first and last good bins are read from the data file into the vectors `IT0`, `IT1` and `IT2`. The vectors `NT0`, `NT1` and `NT2` alow the user to override the file specifications. These parameters can conveniently be set explicitly with the `BINS` and `TIMES` commands, and automatically with the `T0` command (described in the next section).

4.4 The μSR Commands (Macros)

The "μSR commands" are defined by `ALIAS`ing them in PLOTDATA.INIT to `EXEC`ution commands for their respective command macro files. The following is a categorized list of these commands.

4.4.1 General purpose commands

INIT This command just calls PLOTDATA.INIT anew, reinitializing all the control parameters, fitting conditions, *etc.* It is nearly as thorough as `QUIT`ting PLOTDATA and starting it up again, but takes much less time.

HLP This command offers VMS-style on-line HELP for these utilities. The command `HELP` is a built-in PLOTDATA command for all the other built-in PLOTDATA commands.

HOW As was mentioned above, `HOW RUN` gives information that describes how the TD-μSR macros are configured, and how the commands will process and plot the data. For a list of the accepted `HOW` commands and the information they provide, enter `HOW HOW`.

4.4.2 Data input command

RUN [*e.g.*, `RUN 5001`] Reads from disk the data corresponding to the given run number.

4.4.3 Commands to set display ranges

BINS [*e.g.*, `BINS 217 800`] The `BINS` command takes two arguments: the first and last bins to be used in any histogram operations following. What the command does is to set the values of `NT1` and `NT2` to those numbers, for all histograms. To reset the bin range selection so that it takes the defaults (`IT1` to `IT2`), just set `BINS 0 0`.

TIME This is just like `BINS` except the arguments are *times* defining a range (in nanoseconds) to be displayed. The `NT1` and `NT2` pointers are set accordingly for each histogram separately. Because this macro depends upon accurate determination of $t = 0$ (either from `IT0` or from `NT0`), it is slightly more fragile than `BINS`, but probably much more useful.

4.4.4 Graphic commands

T0 This command causes the region near `NT0` and markers for both `NT0` and `IT0` to be displayed for each Histogram, Fig. 4.1. `NT0` is the bin number of time zero as determined by PLOTDATA and `IT0` is the number read from the data

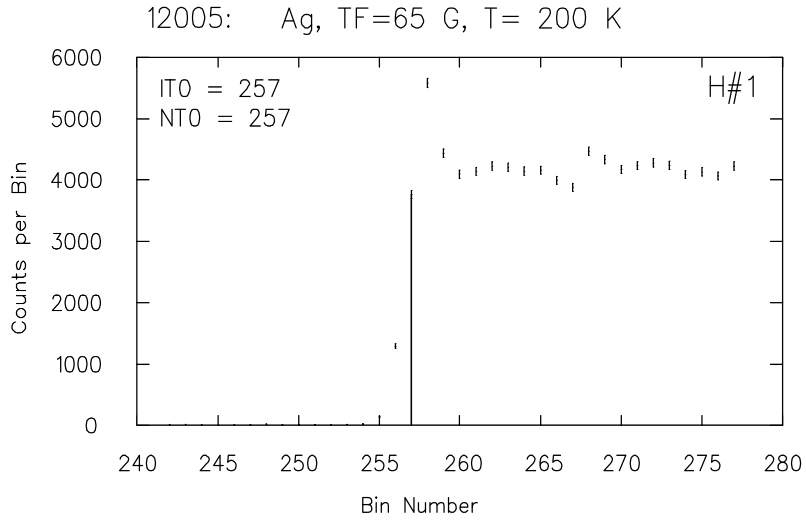


Figure 4.1: The output of the `T0` macro is the region near `NT0`. Shown in this figure is the output for one histogram. Normally the screen is divided into windows and the region near `NT0` is shown for all histograms.

file. While setting up, the user should determine `NT0` and enter the correct time zero, first and last good bin into the data file (using the `Modify...  Mode... Modify Mode Settings` command of the TD-muSR data acquisition program).

BKGD Determines the background bin range and confirms visually that the region in the $\mathtt{j}^{\mathrm{th}}$ Histogram estimated to contain flat $t < 0$ background (bins `IBG1[j]` to `IBG2[j]`) actually does, Fig. 4.2.

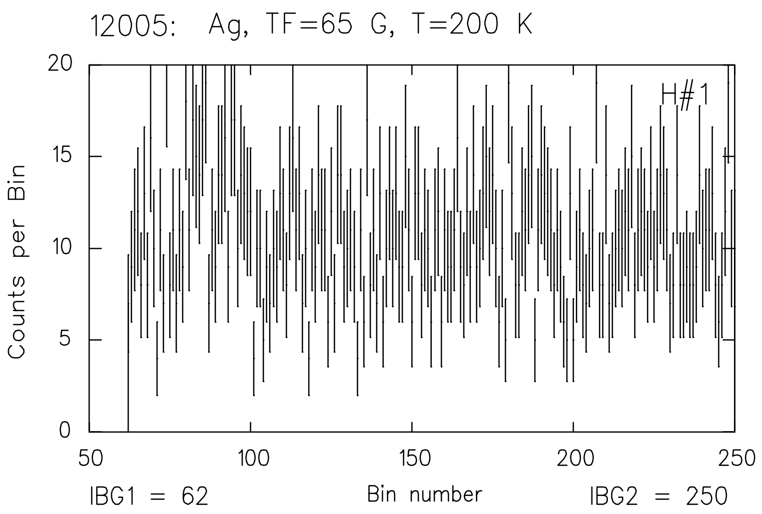


Figure 4.2: Typical output from the `BKGD` macro.

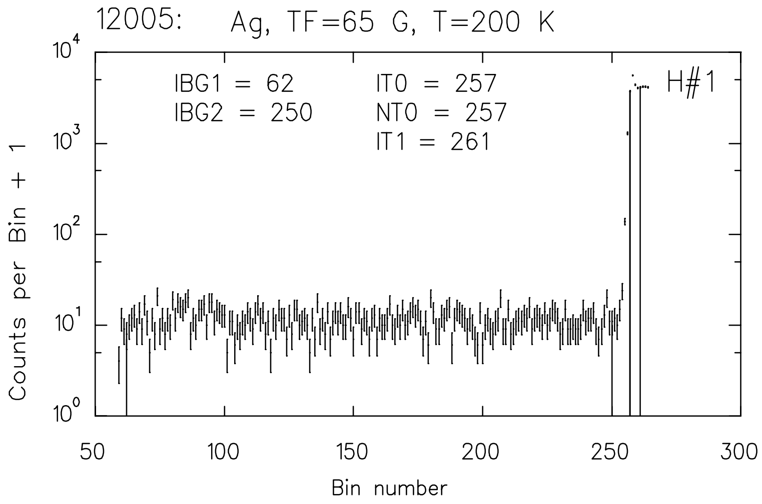


Figure 4.3: Output of the `EARLY` macro.

EARLY This command is a sort of hybrid of `T0` and `BKGD` in that it displays the region from just before `IBG1` to just after `NT1` [or `IT1`] for each histogram and displays the values of `IBG1`, `IBG2`, `NT0`, `IT0`, `NT1` and `IT1` on the plot (colour-coded if you have a colour display); Fig. 4.3. The vertical axis is shown on a log scale so that background and signal will both show up clearly. The `EARLY` command will only determine new `NT0` values if the current `NT0` values are zero, so if you adjust the experiment's timing, you must either execute `T0` explicitly or zero the `NT0` array before executing `EARLY`.

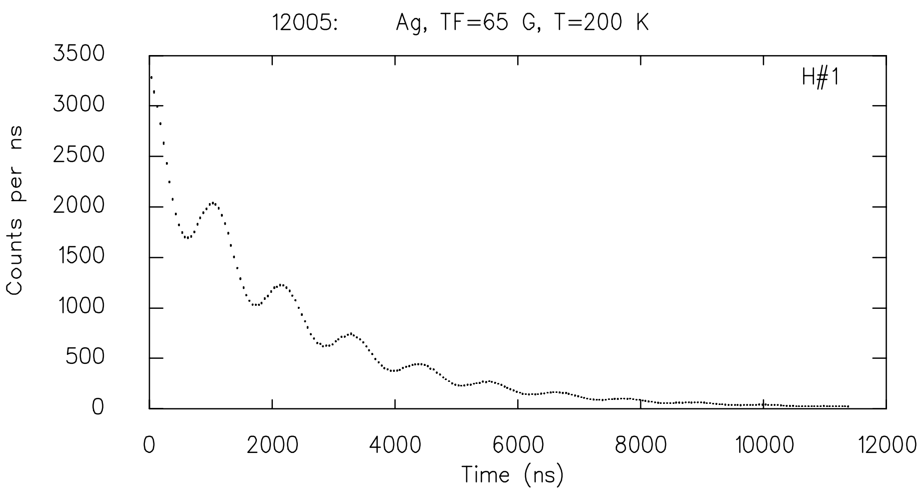


Figure 4.4: Output of the RAW macro. Also, the data can be displayed as a function of bin number instead of time.

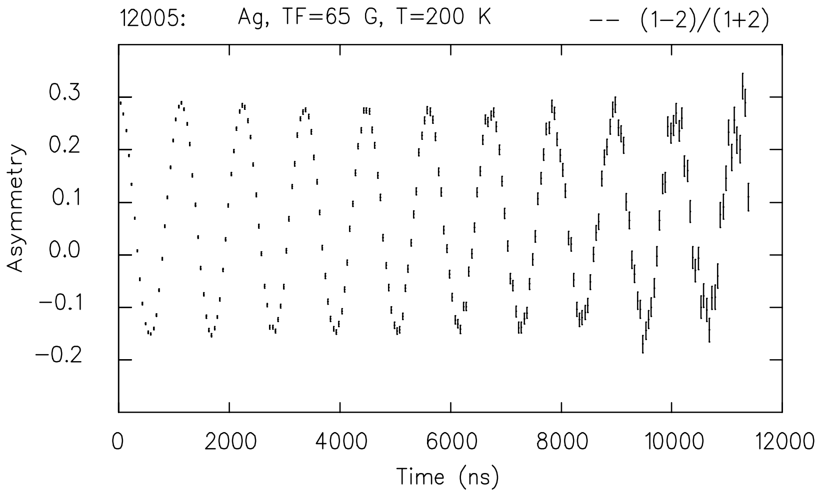


Figure 4.5: Output of the ASY macro $A(t) = [1-2]\ /\ [1+2]$, for `JHST={1,2}`.

RAW This command will display the raw histogram data displayed as a function of *bin number* $[N(i)]$ or as a function of *time* (ns) $[N(t)]$, Fig. 4.4.

If you choose the former, it displays, for each $[j^{\text{th}}]$ Histogram, the raw $N(i)$ data starting at `NT1`$[j]$ if it is nonzero or `IT1`$[j]$ otherwise, packed by `NPAC` to Bin `NT2`$[j]$ (`IT2`$[j]$) or 256 packed bins, whichever comes first. (It won't let you waste your time waiting for display of 8,000 bins of raw data!) If `NPAC=0`, it will display *unpacked* data in this mode.

If you choose $N(t)$ mode, you get the same thing except (a) the spectra are displayed as a function of t [ns]; and (b) if `NPAC=0`, a packing factor of `NB/256` (where `NB` is the number of raw bins in the selected range) is used to produce 256 bins of packed data for the display. That is, the whole selected time range is displayed in the most economical fashion.

If you don't want to see all `NHISTS` Histograms, you can set `NHMAX=1` to see just the first; if you want to see only Histogram 4, set `JHST = {4,1,2,3}` (or whatever) as well as `NHMAX=1`. (But don't forget you have done this; it stays set that way until you change it!) The macro automatically figures out how to display the required number of Histograms on one screen.

ASY This command is like `RAW` [in $N(t)$ mode] except that it displays $A(t) \equiv [F(t){-}B(t)]\ /\ [F(t){+}B(t)]$ for each *pair* of Histograms $F(t)$ and $B(t)$, which first are aligned to a common $t = 0$ and background-subtracted, Fig 4.5. As in `RAW`, no more than 256 packed bins will be displayed; but here `NPAC=0` will always be treated as in the $N(t)$ mode of `RAW`.

If there is only one histogram, then we use the numerical algorithm appropriate to a high-frequency signal and a μ^+ lifetime to extract $A(t)$. If there are (e.g.) 3 Histograms, we assume that the first 2 form a "F-B" pair and the third is a

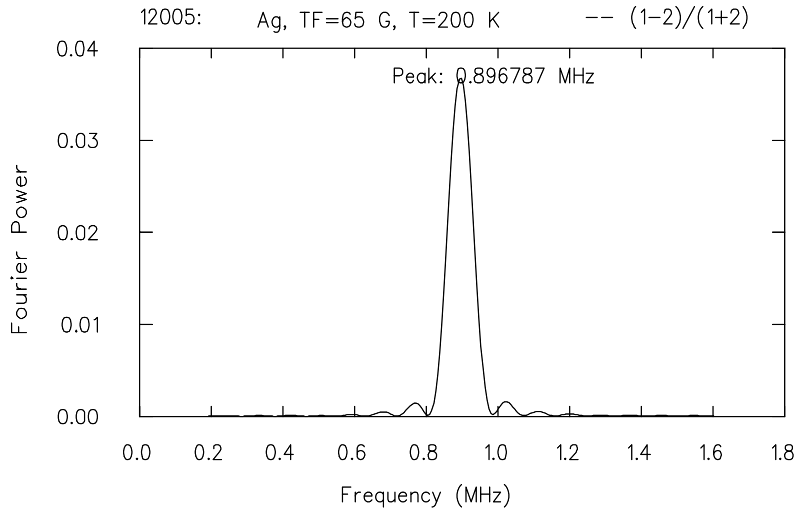


Figure 4.6: Typical output of the FRQ macro.

"loner"—and so on. You can always get what you want by manipulating `NHMAX` and `JHST`, but any sensible configuration should come out the way you want it automatically.

FRQ This would be called `FFT` were it not for the fact that `FFT` is a native PLOTDATA command. This forms $A(t)$ as in `ASY`, removes any constant "baseline" and performs a Fast Fourier Transform. The resultant frequency spectrum is plotted between the limits `FRQ1` and `FRQ2`, if they are nonzero, or over the ± 64 frequency bins centered on the peak frequency `PKFRQ` if `FRQ1` or `FRQ2` is zero Fig. 4.6. Only the first 1 or 2 Histograms are used in the FFT.

TILE [*e.g.*, `TILE ASY 4998 5001`] The `TILE` command takes 3 arguments: the *type* of plot and the *range* (first, last) of *run numbers* for which to juxtapose said plots.

The idea of `TILE` (or `STILE`—see below) is to allow several runs to be plotted next to each other on the same "page". The screen is automatically split up into as many `WINDOW`s as it takes to show all the specified runs at once.

Advanced users will learn to set `AUTOSCALE OFF` after a particularly attractive single plot and before making a `TILE` plot, so that all the runs get plotted with the same vertical scale. Don't forget to set `AUTOSCALE ON` when you're done!

STILE [*e.g.*, `STILE ASY`] The `STILE` command is just like `TILE` except that it takes only one argument: the *type* of plot. The list of runs is stored in the file STILE_FILE.LST, created by the `SRUNS` command (or by editing, if you prefer); it can contain runs in *any sequence*, which makes it handy when your runs are not all taken in the order you want them displayed. The format of STILE_FILE.LST is as follows: the first line gives the number of runs and each subsequent line contains a run number. For example,

```
4
5701
5711
5707
5702
```

4.5 PLOTDATA commands

This section describes a few PLOTDATA commands that are often used in addition to the μSR commands are described. For a full description of PLOTDATA, see the manuals *"PLOTDATA Users' Guide"* and *"PLOTDATA Command Reference Manual"*.

HELP Invokes the online PLOTDATA help facility.

QUIT Exit and unload PLOTDATA.

CLEAR Is used to clear the graphic display.

HARDCOPY Produce a hardcopy of the graphic display on some specified printer.

AUTOSCALE This command controls the autoscaling of graphic axes. The most used commands are `AUTOSCALE ON` and `AUTOSCALE OFF`.

SCALES x_1 x_2 n_x y_1 y_2 n_y This command turns off the autoscaling and sets the scales as follows:
x_1: minimum value to display on the x-axis.
x_2: maximum value to display on the x-axis.
n_x: number of numbered tic marks on the x-axis.
y_1: minimum value to display on the y-axis.
y_2: maximum value to display on the y-axis.
n_y: number of numbered tic marks on the y-axis.

SET a great many PLOTDATA parameters [*e.g.*, `SET YLOG 10` Sets the y-axis to logarithmic scale base 10]. See the PLOTDATA manual for lists of PLOTDATA parameters.

μSR PLOTDATA Quick Reference

Run File

Command [*e.g.*, RUN 5001]

RUN	Load run #

File specified run information [*e.g.*, =IRUN]:

IH	contains the data; matrix.
IRUN	is the run number; scalar.
IT0	is the bin number where $t = 0$ occurs; vector.
IT1	is the bin number where good data starts; vector.
IT2	is the bin number where good data ends; vector.
NBINS	is the number of bin per histogram; scalar.
NHISTS	is the number of histogram; scalar.
NS_BIN	is the time (ns) per bin; scalar.
TITLE	is the run title; text string.

Display Control

Scalars [*e.g.*, NPAC=10]

FRQ1	Lower limit on FRQ frequency range (MHz).
FRQ2	Upper limit.
NHMAX	Maximum number of Histograms to process.
NPAC	Packing factor (0: PLOTDATA will choose the packing).

Vectors [*e.g.*, JHST={1,4,2,3}]

IBG1	First bin of t< 0 background.
IBG2	ditto – last bin.
JHST	Correct sequence of Histograms.
NT0	Other than 0 will override $t = 0$ bin specification (IT0) in file.
NT1	ditto – first good bin.
NT2	ditto – last good bin.

Commands [e.g. BINS 217 1200]

BINS	First and last bins in any following operation.
TIME	ditto – arguments are times (ns).

Graphic Commands [*e.g.*, ASY]

ASY	The asymmetry spectrum.
BKGD	The $t < 0$ background.
EARLY	Early times (background to 1st good bin).
FRQ	Fast Fourier transform.
RAW	Raw data.
T0	Determine $t = 0$ and display the region near it.

Display more than one run [see text].

TILE	Display a range of runs.
STILE	Display many runs.

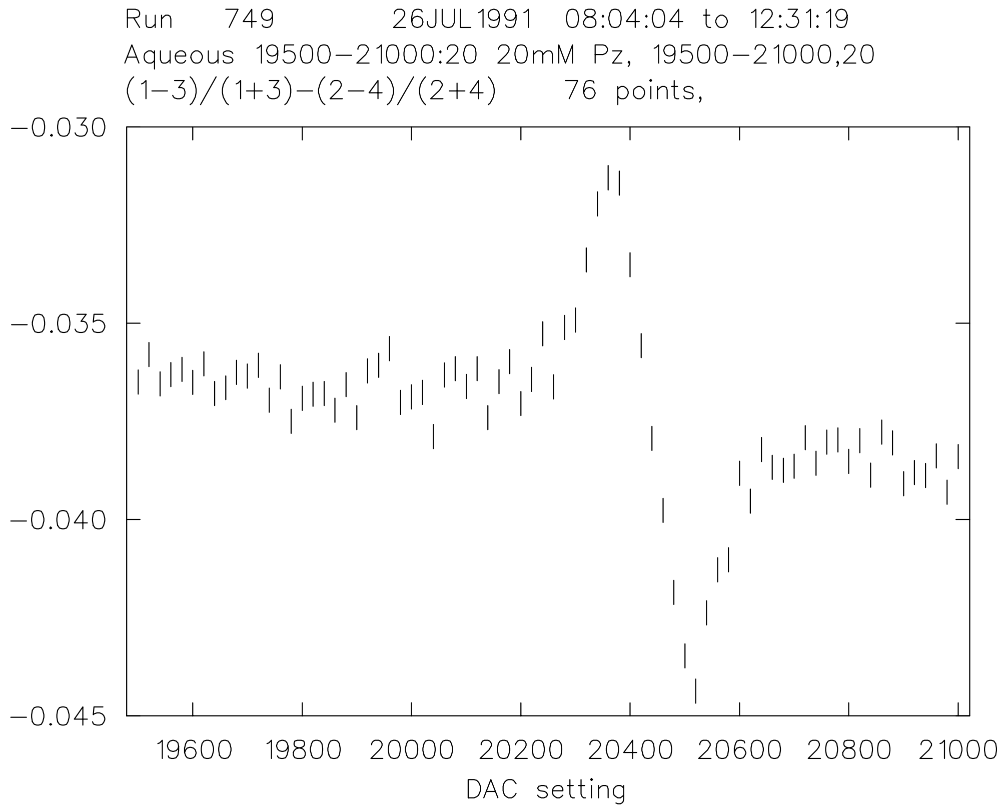
General purpose commands

HLP	Invokes the help facility.
HOW HOW	Another source of help.
INIT	Reinitialize all control parameters.

Few PLOTDATA commands

AUTOSCALE ON	Controls the autoscaling of axis.
AUTOSCALE OFF	Turns autoscaling off.
CLEAR	Clear Graphic display.
HARDCOPY	Obtain a hard copy of the graphic display.
HELP	Invokes PLOTDATA help facility.
SCALE	Sets the graphic scales.
SET YLOG 10	Sets the y-axis to logarithmic scale base 10.
QUIT	Exit PLOTDATA.

5 LCRplot: Displaying Time-Integral Data



LCRplot is a program for online display of integral-μSR runs.Its aim is to produce useful graphs, rather than the prettiest ones, with little effort and no waiting. LCRplot is specialized for plotting asymmetry (calculated in various "modes") *vs.* magnetic field (expressed in "DAC units" or mA of current) for one or several I-μSR runs. To use LCRplot when taking data on the beamline data acquisition computers, simply type

```
lcrplot
```

From the main data analysis cluster you should type

```
run usr0:[msrplt.lcr]lcrplot
```

5.1 LCRplot commands

LCRplot uses a simple command interface; the commands are few and there are only two distinct contexts for the commands: the main command prompt and the help prompt. In addition, LCRplot will prompt you for specific information when necessary. The commands are case-iNsEnsitIve, and they all single letters (but this may change if and when new commands are added). Table 5.1 lists all the LCRplot commands.

Table 5.1: LCRplot command summary

?	print a quick-reference menu, like this
number	plot different run, as given by run number; autoscale x axis
blank	replot (preserve x- and y-axis ranges if they were explicit)
H	help for each command, and some related concepts
R	replot; switch to autoscaling both axes
D	replot using specified DAC range and bin size for x-axis; autoscale if no range given
M	change plot mode and mapping of scalers
N	plot from one to twenty runs, preserving axis ranges
L	locate runs in different directory
S	subtract linear or polynomial 'background'
U	update plot of current run at regular intervals
P, C	print a hard copy on Laserjet, Printonix, or PostScript
W	write points to file for a fancy plotting program
X	replot points in order of acquisition
Y	replot with new y-axis boundaries (default = autoscale)
K	delete (kill) some data points
E, Q	exit, quit

The axis scales are generally controlled by the `D`, `X`, and `Y` commands. `Y` controls the y (asymmetry) axis, while `D` and `X` control the x axis; but you should almost always use the `D` command which scales the axis in units of magnet current (mA) or DAC value (there is no field calibration in this program). When parameters are omitted from these commands, automatic ranging is used. The `R`, `L`, and ⟨*run_number*⟩ commands also cause autoscaling (the `N` command will get a new run with the same scaling).

Examples:

`d 1000,1500,25`	(averages points into bins 25 DAC units wide)
`D 14000,18000`	(plots every point within the given DAC range)
`D ,,1`	(averages all points that have identical DAC)
`Y -.001,.001`	(zooms in on an asymmetry range around zero)

See the descriptions of the individual commands for more detail.

In the command descriptions which follow, optional parameters are given in square brackets, " [ ] ", characters you should type exactly are given in `typewriter` or **bold** face, and numeric parameters are represented by *italic* letters. The commands themselves are shown in capital letters, but they can be typed in lowercase. As mentioned, most commands are a single letter, but the individual descriptions will begin with ones that are *not* letters, and then move through the alphabet.

5.1.1 Blank or Null

This is the 'Press Return' command which refreshes the plot. Pressing the return key at the `LCR>` prompt causes the current plot to be replotted using the same x- and y-axis ranges, if any have been specified, or using autoscaling if autoscaling was used previously. This is useful when a run is still in progress.

See also the `N` and `U` commands.

5.1.2 Run Number

n

Entering just a number, without any other command, selects a new run to plot and sets the x axis to autoscale. If automatic updates are in progress, they are stopped. If the specified run does not exist or is not an LCR run, you will be prompted for a new run number. It may be that a run can't be found because it is in some unknown directory; you should then use the `L` command to specify the location.

Example: `911`

To select a new run but preserve the current axis scaling, use the `N` command.

5.1.3 ? – quick help

The `?` command displays the quick reference list for LCRplot commands. You will be returned to the `LCR>` prompt immediately.

5.1.4 C – hardCopy

C [d]
P [d]

Both the `C` ('Copy') and the `P` ('Print') commands do the same thing: they print a hardcopy of the current plot on device number d. If d is omitted, the following list of legal devices will be shown and you will be prompted to specify one.

Code	Printer	Bitmap	Pages	Ori
1	Printronix	700×785	1	L
2	HP LaserJet	985×750, 100 dpi	1	L
3	HP LaserJet	1120×850, 150 dpi	2/3	P
4	HP LaserJet	1128×846, 300 dpi	1/6	P
5	Postscript	N/A	1	L

In either case, you will be prompted for what to do with the hardcopy; the most common answer is '`P`' (print). If, however, you want to combine plots (perhaps with text) using the RPrint utility (`[msrplt]rprint.com`) specify code 3 or 4, and print option `T` (TEX).

Examples:
```
p
C 2
```

5.1.5 D – DAC range

D [m, n] [$, b$]

Set x-axis range in terms of DAC values (or magnet current). Replot, selecting points with DAC values from m to n, and group the points into bins b DAC units wide.

If no parameters are given, all points are plotted without any binning. If just the range (m, n) is given then just the points between m and n (inclusive) are plotted, without any binning. Multiple scans will print points over each other, distinguished by color only: green for scans with increasing DAC and violet for decreasing DAC.

If all parameters are given, then points from m to n are selected, and averaged into bins b DAC units wide. the bin size can be as small as 1, in which case only points with the same DAC value are averaged together. If m and n are absent but b is given, all points are selected, but grouped into bins of size b. A very useful command is `D ,,1` which combines multiple scans by averaging all points with the same DAC values.

If $m > n$, all parameters are ignored. Remember, 'b' is a bin *size*, not a packing factor.

Examples:	`D`	(plot each and every point on a DAC scale)
	`D ,,1`	(average all points that have identical DAC)
	`D ,,30`	(average points into bins 30 DAC units wide)
	`d 100,2500,25`	(plot points in range 100–2500 and bin them)
	`D 14000,18000`	(plot every point within the range 14000–18000)

See also: `X`, `Y`.

5.1.6 E – Exit

E
Q

The `E` (Exit) and `Q` (Quit) commands are identical. They terminate the LCRplot program.

Examples:	`E`
	`q`

5.1.7 H – Help

H [*command*]

Get help on the desired command. If no command is given, LCRplot displays a quick reference list and prompts you for the name of a command. To leave the help facility, press the 'Return' key at the help prompt.

Examples:	`H`
	`h u`

5.1.8 K – Kill bad data points

K

The Kill command is a preliminary version of what might become a more capable data editor, but at the moment all it can do is delete.[1] Upon receiving a `K` command, LCRplot replots the current run(s) with autoscaling and no binning, and then enters a visual editor mode. In the editor, a colored flashing cursor marks where you are editing; various keys move the cursor and/or delete points, as listed in Table 5.2 on the following page.

[1] This may be the best method for the future too—it prevents the overzealous and imaginitive editing that would surely occur if one could shift sections of data for removing steps.

Table 5.2: LCRplot "K" command editing keys

Key	Function
→ (←) cursors	Move to the next (previous) data point
↑ (↓) cursors	Move to the same DAC value in the next (previous) sweep
N, Next	Move cursor to the beginning of the next sweep
P, Previous	Move to the end of the previous sweep
D, K	Delete the current data point
Del	Delete the current point and move to the previous point
Space	Delete the current point and move to the next
R, Remove	Delete the entire sweep containing the current point
I, Insert	Restore all points in the current sweep
H, ?	Display a table like this one
U	Apply changes, refresh the plot, and continue editing
Q, Enter, Do	Apply changes and stop the killing

The cursor movements might prove confusing, particularly in a plot *vs.* DAC setting. The terms "next point" and "previous point" refer to the order of acquisition, not to the DAC value, so in sweeps of decreasing DAC (violet points) the right cursor key (→) will move the cursor to the *left* on the screen. Similarly, the up and down keys have nothing to do with up or down on the screen, although they do preserve the horizontal position on a DAC plot. It is probably easiest to edit a plot using the "X" range rather than DAC. The range and mode can be changed after editing.

Deleted points are removed from internal arrays; they are *not* removed from the data file! All deletions will be restored when a new run is selected or the "R"eplot command is given.

5.1.9 L – Locate data files

L

Use the `L` command to specify the directory containing LCR data files. The default is to use the current year's runs in the standard archival directory, so the `L` command is necessary to access previous years' runs or current runs on a different beamline. You are prompted for the new directory; there are no command-line parameters.

If you enter a null response to the prompt, you will be prompted once again. If you again enter nothing, the current default directory is checked first, followed by the archive directory appropriate for the specific computer. A response of `^Z` leaves the directory unchanged.

It is possible to enter descriptive filename formats for more flexibility. To use this, enter a text string containing "%n" where n is a single-digit number (0–9). The filename chosen will be that string with the "%n" replaced by the run number, padded with zeros out to 'n' digits unless $n = 0$ where the natural length for the run number is used. Two examples illustrate:

On the data acquisition computers, data files are renamed at the start of a new year. Suppose it is January 1997 (it is when writing this) and you want to see run 6488 from 1996 on the M15 beamline; enter the location

`IMUSRED:[DLOG]%0.DAT_1996` $\longrightarrow$ `IMUSRED:[DLOG]6488.DAT_1996`

Now suppose run 321 from M20 gets archived incorrectly on the cluster as a TD run. Its location and name format could be

`MSR5:[M20.1997.OLD]%6.TRI` $\longrightarrow$ `MSR5:[M20.1997.OLD]000321.TRI`

The directory name is remembered in the logical name DDLCR, and you can set this before running LCRplot, in which case the `L` command is unnecessary. DDLCR is also remembered from one run of LCRplot to the next, provided you do not log off. Whenever DDLCR is undefined, the current default directory is checked first, followed by standard repositories.

Automatic updates are canceled if they were active.

5.1.10 M – Mode

M [m [$, i, j, \ldots$]]

How LCRplot calculates asymmetry from scaler values is set by the **M**ode command, specifying the 'mode' setting (m) and a choice of scalers (i, j, *etc.*) You can enter all numbers on one command line, or just `M` and answer the prompts. The modes available are

1: A single scaler; 'A'.

2: Two scaler asymmetry; $(A-B)/(A+B)$

3: Ratio of two scalers; A/B

4: Four scaler differential asymmetry; $(A-B)/(A+B) - (C-D)/(C+D)$

5: Two scaler asymmetry with $\alpha\,\beta$ correction; $(\alpha A-B)/(\alpha\beta A+B)$

6: Four scaler differential asymmetry with $\alpha\,\beta$ correction; $(\alpha A-B)/(\alpha\beta A+B) - (\alpha C-D)/(\alpha\beta C+D)$

The canonical assignment of scalers is: $A = \mathrm{B}^+ = 1$, $B = \mathrm{B}^- = 2$, $C = \mathrm{F}^+ = 3$, $D = \mathrm{F}^- = 4$, but these can vary depending on the set-up. The default mode is mode 4, with these scaler assignments, appropriate for an experiment with a flip coil, and giving the differential asymmetry $(\mathrm{B}^+-\mathrm{F}^+)/(\mathrm{B}^++\mathrm{F}^+) - (\mathrm{B}^--\mathrm{F}^-)/(\mathrm{B}^-+\mathrm{F}^-)$; that is, `M 4, 1,3,2,4`. You may enter a different mode upon start-up when LCRplot prompts you, or you can enter the `M` command any other time.

The Mode command sets the y-axis range to autoscale because different modes need different ranges.

A handy trick: If you have 4 scalers but want a non-differential plot, you can *add* the + and − modulation instead of subtracting them by using the command `M 4, 1,3,4,2`, whereas the regular differential ($\mathcal{A}^+ - \mathcal{A}^-$) plot is given by `M 4, 1,3,2,4`.

Any asymmetry can be flipped by reversing the A, B assignments; compare `M 2, 1,3` with `M 2, 3,1`. For mode 4, reverse both $A \leftrightarrow B$ and $C \leftrightarrow D$.

Another method for differential asymmetry is to calculate a modulation-asymmetry for each direction, then take the difference between the directions: $(\mathrm{B}^+ - \mathrm{B}^-)/(\mathrm{B}^+ + \mathrm{B}^-) - (\mathrm{F}^+ - \mathrm{F}^-)/(\mathrm{F}^+ + \mathrm{F}^-)$, selected by `M 4, 1,2,3,4`. This is best for integral-RF runs where the "−" modulation has the RF off for no signal; it is a good approximation to the asymmetry corrected for α.

Modes 5 and 6 have correction factors to account for experimental artifacts such as counter geometry and efficiency, and they are rarely used for online run display. The ratio of counting efficiencies is $\alpha \equiv \epsilon_\mathrm{F}/\epsilon_\mathrm{B}$ and the ratio of muon asymmetry measured in each counter is $\beta \equiv Asy_\mathrm{F}/Asy_\mathrm{B}$. Typically, the α parameter changes a lot as the magnetic field is varied, so a single determination is not useful for an I-μSR run. The β parameter must be determined with TD runs and is usually left at $\beta = 1$.

After reading the scaler assignments, LCRplot prompts for three parameters: α_0, α-slope, and β; or the special key "`A`" for "automatic". These provide for a linearly varying α value with a fixed β (usually 1). Any parameters left blank retain their previous values, and the default values are $1, 0, 1$ (no corrections). Automatic entry means that LCRplot calculates the appropriate ratio for the mode—B/A for mode 5 and $(B{+}D)/(A{+}C)$ for mode 6—and fits a straight line to it. These ratios give α for a run with either a steady sweep (5) or flip-coil modulation (6), provided there is little or no signal. After setting the parameters based on a run with no signal, you can switch to another run with the same baseline but a large signal.

For RF runs, α calibration is not necessary since the proper mapping of scalers provides good correction with mode 4.

Examples:	`M`	(Prompt for everything)
	`m 2`	(Asymmetry; will prompt for scaler mapping)
	`m 2, 3,1`	(Asymmetry; typical scaler pair)
	`M 4, 2,4,1,3`	(Differential asymmetry)
	`M 4, 1,3,4,2`	(Sum of asymmetry in + and − modulation)
	`M 4, 1,2,3,4`	(Asymmetry for RF modulation)

5.1.11 N – Number of runs

N $i\,[,j,k,\ldots]$

Select one or more runs $i, j, k\ldots$, collecting them all into one plot, using the previously-specified axis ranges. Automatic updates are canceled if they were active. The `N` command is useful even for single runs because the old axis scaling is retained, whereas typing the run number without any command forces autoscaling.

Autoscaling of the DAC-axis ('`D`') works fine, but if auto-ranging is used with binning (*e.g.*, '`D ,,20`') the range is determined by the first run alone.

A handy trick: If you take a series of runs with small DAC offsets from one another, it is much easier to show them in a single plot. For example, two runs with the sweep ranges 12000–13000:20 and 12001–13001:20, can be displayed well together using the scale `D ,,1` — each run will have its several scans averaged, but no average will be taken between the runs. To deliberately take the average of these two runs you can use the range `D ,,20`.

Examples:

```
N 123
N 5112,5115,5118,5133,5134
```

To plot a single run with auto-DAC-scaling, just type the run number with no `N` command.

5.1.12 P – Printout

The `P` command is the same as the `C` command.

5.1.13 Q – Quit

Q
E

The `E` (Exit) and `Q` (Quit) commands are identical. They terminate the LCRplot program.

5.1.14 R – Replot

R

Replot the current run autoscaling both axes. The `R` command has the same effect as retyping the current run number or issuing the pair of commands `D` and `Y`.

5.1.15 S – Subtract

S
S S
S P [*i*]

Subtract background slope or background polynomial function from data. (Such subtraction is unnecessary when using the differential mode 4.) Typing `S` alone toggles (on/off) subtraction of the linear component of the data. When slope subtraction is enabled, the data is fit with a linear regression, and the straight line fit is subtracted before plotting. This makes it easier to see resonances when the data is strongly sloped (due to magnet focusing).

The command `S S` fits the slope for the current data and saves those values for subsequent plots; even for other run files. It is good to issue the `S S` command when plotting a run with little signal, and keep the same subtraction when plotting other runs.

The `S P` command allows you to subtract a specified polynomial function. Normally, the polynomial is a fit to a background run (*e.g.*, using plotdata; see the notes on fitting below). The optional parameter i is the degree of the polynomial; use 0 to turn off polynomial subtraction. LCRplot prompts for $i+1$ coefficients. These are NOT the simple polynomial coefficients, but are $C_0, C_1, \ldots, C_i$ in:

$$C_0 + C_1\,x + \frac{C_2}{|C_2|}(C_2\,x)^2 + (C_3\,x)^3 + \frac{C_4}{|C_4|}(C_4\,x)^4 + \cdots$$

where $C_2/|C_2|$ is just ± 1 retaining the sign of the coefficient. This parameterization avoids numeric overflows. Make sure you use the right function when fitting!

Examples:	`s`	(toggle [on/off] slope subtraction)
	`s s`	(turn on slope subtraction and retain slope)
	`S p 5`	(subtract a 5th order polynomial)
	`s p 0`	(turn off polynomial subtraction)

5.1.16 U – automatic Updates

U [*i*]

The update command only works while data acquisition is in progress. Whichever run is in progress will be plotted, and replotted as new points are added to the file. Moreover, the automatic updating will switch to new runs as they are started. If you have been examining runs from a different directory (after an `L` command) the data directory will automatically change to the online acquision directory `imusred:[dlog]`.

To stop the automatic updates enter a run number or another `U` command at the `LCR>` prompt.

If the the optional parameter is given, the update interval is changed to 'i' seconds and updates are not cancelled. When updates are in progress, a screen message reports this fact.

Examples: `U`
`U 60`

5.1.17 W – Write data file

W [F]

Write the plot points out to a file using format F (where F is a letter). The file will serve as input for another plotting program:

F	program	filename
D	db	*nnnn*.db
P	Plotdata	*nnnn*.PLD
X	XYplot	*nnnn*.XYP
Z	Zplot	*nnnn*.Z

where *nnnn* is the current run number. Note that points outside the y-range are included in the file, but not points outside the x-range.

Examples: `W`
`w db`

5.1.18 X – x-range (in counted points)

X [m, n] [$, p$]

Set x-axis range in terms of counting points: Replot from point m to point n packing in groups of p points. If m and n are not given, all points are plotted. If p is not given, it defaults to the lowest value that gives no more than 256 plot points (usually 1). Thus, if no parameters are given, all points are plotted, but with some packing if there are very many points. If $m > n$, all parameters are ignored.

This type of scaling is hardly ever used. Use the `D` command instead.

Examples:
```
X
X 1,300,5
x ,,2
X 20,30
```

5.1.19 Y – set y-axis (asymmetry) range

Y $[r, s]$

Set a range for the vertical axis. If the range r, s is omitted, or if $r > s$, the graph is autoscaled to fit the points. Note that the y-axis range is preserved when changing runs, but lost when changing modes.

Examples:
```
Y
Y -.01,.01
```

5.2 Special Keys

Full command line editing and re-entry is available, but only at the `LCR>` and help prompts. For a list of the editing keys, and all special-purpose keys, press PF2. To activate a calculator, press PF4.

5.3 Fitting

Eventually, LCRplot should be expanded to do run fitting, but right now there is no fitting capability. If you want to subtract background runs, you will need to fit them to a polynomial, and you should probably use plotdata for that. Here's how:

First, read the points into plotdata, either via a file made by LCRplot (`W` command) or directly with RESTORE/IMUSR; *e.g.*,
```
READ 1234.PLD D Y DY
W=1/(DY*DY)
```
Then, for a 5th order polynomial, declare C_0 through C_5 using
```
VARY C0
C0=.001
...
```
and fit them with
```
 FIT/WEIGHT W  Y=C0+C1*X+(C2*X)**2*C2/ABS(C2)+(C3*X)**3+
   (C4*X)**4*C4/ABS(C4)+(C5*X)**5
```
Note the `C/ABS(C)` factors in the even terms! They give the sign of the term which would be lost otherwise. The fitted coefficients can then be used by LCRplot for background subtraction.