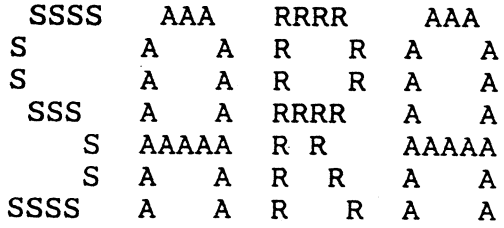
[POLLACK.USERDOCS]SARA.WM

MSUCL report no. 499

CREATED 4/03/83
REVISED 2/21/87



THE SARA DATA ANALYSIS PROGRAM

****

Authors: Brad Sherrill, John Winfield

Contributors: Ron Fox, Mark Lowe

National Superconducting Cyclotron Laboratory
Michigan State University
East Lansing
MI 48824-1321

TABLE OF CONTENTS

I. INTRODUCTION

This document describes the operation and use of the SARA histogramming and data analysis program at NSCL. The code is written in VAX FORTRAN-77 and source files may be found on the ANALYS machine under the directory [SARA]. On-line data taking is performed in conjunction with the ROUTER system of programs and is specifically dealt with in Section II. Off-line analysis is available in association with the tape-reading program LOG as described in Section IV. Section III is general.

II. DATA ACQUISITION.

II(a). Starting from Scratch.

To begin one should be logged on under the appropriate experiment account number (see a member of the computer group if it doesn't exist). If you are starting from scratch, make sure you have a file called LOGIN.COM on the account (even if it is an empty file), and type

```
@[SARA]INITSARA
```

to initialize some command files for starting up the data acquisition processes (this only has to be done once on a given account). This appends an appropriate command file from [SARA] to the user's LOGIN.COM -- you will be asked which data acquisition system you intend to use.

INITSARA will also copy a file containing a subroutine DATAR, which is the user-definable piece of SARA. The file copied is a skeleton which has no software-generated parameters ("pseudo" parameters). Section III(a) explains how to modify DATAR if necessary, for your experiment.

II(b). Determining On-Line Parameters.

The data acquisition code (including scaler set-up, etc.) should be built at this stage. See the relevant documentation (DAQ Program Generator Guide, R. Fox and A. Grover; Data Acquisition Commands, R. Fox; Using the 68000 Data Acquisition System, A. vander Molen et al.) for instructions.

II(c). Initial Setup of SARA files.

Once your on-line parameters are finalised, a definition file for the histogramming program, i.e. SARA, can be made. Section III(b) describes the format of this file. If pseudo parameters are required, the subroutine DATAR should be edited before it is compiled and linked to the rest of SARA (see Sec. III(a)), so that you have an executable file SARA.EXE on your account.

II(d). Starting the Data Acquisition Processes.

To start up the data acquisition process and SARA, type:

@GOGOGO

You will be asked whether you wish to start up the tape, scaler and histogramming processes at this stage. If you are not ready to begin any of these, they can be started up at a later time by individual commands (do not type @GOGOGO again). Note that the tape process must be running before any data may be written to tape, regardless of the status of the "TAPE ON/OFF" switch.

The processes NOTIFIER and ROUTER should be running after GOGOGO.COM has completed. Check this by typing

SHOW PROCESS/SUB

If they are not present, some error has occurred (CAMAC crate not powered up or not on-line, for example), and you must correct the fault and begin again.

II(e). Sampling of Buffers.

As written, the command file GOGOGO.COM will cause data buffers to be sampled by the histogramming program (i.e. on-line data is only sent to SARA when the computer isn't busy doing other things such as writing data to tape or accumulating scaler records). Control buffers (begin-of-run, end-of-run, pause, resume) are sent to SARA in the "no-sample" mode, so SARA should always know the run status (unless it has been restarted in the middle of a data-taking run). Scaler buffers are not sent to SARA in the on-line mode. These choices of what buffers are and are not sent to SARA may be changed by editing GOGOGO.COM if desired.

In order to estimate the amount of data actually acquired on tape, one needs to know the sampling efficiency. This is determined from an accumulated count of the number of buffers received by the histogramming program compared to the sequence number of the buffer as assigned by the data acquisition microprocessor. The sampling efficiency may be obtained from the STATus command (Sec. III(f)).

II(f). If SARA crashes during data-taking.

The golden rule is DON'T PANIC ! Data will continue to be written to tape and the scalers will still be updated. Only on-line histogramming is interupted. So... there is no need to stop the current run (although you may do so if you wish). To get SARA going again, type

RESTART

and, when SARA prompts you, enter the name of the most up-to-date definition file. If you're not sure which one this is, you might type

DIR/DATE=MODIFIED *.DEF

before restarting SARA. The file FAILSAFE.DEF should always contain all the histograms and conditions which were defined before the crash, since it is updated every time new ones are created.

Unless you know that the ordering of histograms in the definition file is very much different to that when SARA crashed, there is no need to zero the AED memory. Then most, if not all, the histograms held before the crash will be restored intact. If some histograms look odd, it is probably because the old arrangement of histograms did not completely match the new setup. It is then best to zero these histograms individually.

SARA will begin to increment histograms again as soon as you see the SARA> prompt. It is not necessary to stop and restart the run.

II(g). If the end of a tape is reached during a data-taking run.

If this happens, a message will appear on both the VAX terminal and the console (DECwriter). For the Mark-II acquisition system, it is recommended that the run is PAUSED when this happens, since the scaler process is not stopped automatically and a high effective dead-time might result otherwise. Stopping the run is not advisable since this will cause the end-of- file to be written at the start of the next tape mounted, which might cause confusion. For the Mark-III acquisition system (no LSI), no commands should be issued to the 68k master, since this might confuse the system. Just be patient.

The tape program will rewind the old tape and unload it. Simply wait until it has finished, then remove the old tape, load the new one and put the drive on-line. Writing of data should recommence immediately.

III. SARA OPERATION (both on- and off-line)

The section of SARA which deals with interpretion of buffers from the data stream, the calculation of pseudo parameters, and the binning of histograms is called DATAR (this is the subroutine name; it is used as the generic name throughout this writeup). The FORTRAN file name is, of course, arbitrary -- but on [SARA] we follow the convention that DATAR.FOR is the version for the obsolete LSI-based acquisition system, DATA68K.FOR is the version for the 68000/LSI-based system, and DATAR86.FOR is the version for the 68000/DR11 system.

DATAR is the user-defined piece of SARA. The rest of SARA, which deals with the definition of histograms, parameters and conditions and their manipulation, is standard for all experiments and is linked from the account [SARA] with DATAR from the user's account.

III(a). Pseudo Parameters.

If your experiment requires pseudo parameters, this section is relevant. To create or alter pseudo parameters, edit the program DATAR.FOR (or whatever) and relink it to SARA. This is conveniently done by typing

PSEUDO

The file containing the subroutine DATAR on [SARA] is a "skeleton" containing no pseudo parameters. The parameters are placed in an array P(NO. OF PARAMETERS). The first N elements are the actual adcs tdcs, etc., where N = no. adcs + no tdcs + no. qdcs. Parameter P(1) is the first ADC channel, P(2), is the second ADC channel and so on. If N is defined as above for a given event type, then P(N+1) is the first available location for the a pseudo parameter of that event type (note that pseudo parameters must come after raw parameters, but they do not have to follow immediately after). The proper range for the pseudo parameters is determined by PNBITS(I) in the setup file described in part III(b). The maximum number of allowed parameters, pseudo plus real, is given by the value of NPMAX in DEFCOMMON.

DATAR also has access to variable parameters which the user can change interactively. These are provided to allow normalization of gains not hardware matched, or could be used to correct for particle flight path differences, or might be used as flags, etc. There are two sets: one real*4 array V(NCMAX), and one integer*4 array IV(NCIMAX). The limits NCMAX and NCIMAX are set in the file [SARA]DEFCOMMON.FOR. The method for changing these variables on-line is given in section III(f).

After the editing is finished, DATAR will be compiled and linked to SARA (assuming the command PSEUDO has been used).

If DATAR is compiled "by hand", be sure to use the /NOCHECK option in the FORTRAN command or else the execution time may be increased by as much as 40%. Also, since the default is /CHECK=OVERFLOW, the program will probably crash if the number of counts in a given channel of a histogram exceeds the word-length. After compiling, type

```
@[SARA]SARALINK
```

(this is done for you if the command PSEUDO was used to edit DATAR). You are then asked for the name of the object file to link to SARA. This is normally DATAR (which is the default response), but may be any valid file name if you wish to rename a particular version.

III(b). Definition File.

You must have a setup file to run SARA. This is used to define the histograms and conditions (or "gates"). A working example to tailor your file after is given in Appendix 1. Each definition file must have the five sections in the order:

```
PARAMETERS
-1
VARIABLES
REAL
-1
INTEGER
-1
HISTOGRAMS
-1
CONDITIONS
-1
```

The -1 delimits the end of the section. See the example file for the proper format, and later in this document for histogram and condition formats.

Every time a new histogram is created or a new condition is added from the AED, a "safety" definition file called FAILSAFE.DEF is updated on your account. This feature is designed as an insurance against program crashes or inadvertant exit from the program. Note that FAILSAFE.DEF may not be completely up-to-date in the sense that it is not immediately revised if histograms or conditions are deleted, or if variables are changed.

III(c). The AED display task.

Histograms are displayed through the AEDTSK subprocess. The AED color display is controlled by the commands outlined in Appendix 2 and detailed in [AU]AEDTSK.DOC. The vocabulary and numerology used in AEDTSK.DOC are used in SARA. For example, condition identification, i.e. a band is condition type 3, as taken from AEDTSK.

III(d). Conditions on Histograms.

This sections describes the use of conditions to be satisfied before a histogram is incremented. There are four types of conditions that can be entered from the AED:

gate -- 1D cut with the defining lines included in the condition

line -- 2D cut with the line and every thing falling below the line meeting the condition

band -- 2D non-linear line with everything on or under the line meeting the condition (it takes 2 bands to make the usual definition of a band).

contour -- 2D region with the defining contour and all points inside meeting the conditions. Contours should be no more than double-valued in the vertical direction, or else misleading results may occur.

To define a condition use the appropriate key on the AED terminal:

g -- for gates (two points define a gate)
l -- for lines (one point defines a line)
b -- for bands (up to 18 points define a band)
c -- for contours (up to 32 points define a contour)
e -- erases the previous point.
z -- abort the whole thing.

Once defined the condition must be named. This is done with the "n" key on the AED keyboard. The correct format is:

C<I2><A20> ,

where: C denotes a condition, <I2> is the condition number, which must be unique, and <A20> is an optional name for user bookkeeping.

Valid examples are:

```
C01
C11 14N delta E vs. E
C04 test contour
```

Invalid examples are:

```
Co1 -- o instead of 0
12 -- does not specify a condition
Ctest -- no number specified
C1 pid -- should be "C01"
```

After naming, the condition is accepted by typing "a" on the AED keyboard. If this is not done the condition will not be entered in the data analysis program. Conditions are treated as inclusive of the defining points.

III(e). Format for histogram definition strings - use of conditions.

Four types of histograms may be defined: I*4 or I*2 one-dimensional and I*2 or L*1 two dimensional.

The specification of the histogram definition line is:

H{hist. #}=P{parameter #}(hist. size 2**n)bits-per-chan[conditn]

The correct format for a histogram definition string is:

```
H<I>=P<I>(<I>)I*4[condition]                 for 1D I*4 histograms,
H<I>=P<I>(<I>)I*2[condition]                 for 1D I*2 histograms,
H<I>=P<I>xP<I>(<I>,<I>)I*2[condition]        for 2D I*2 histograms.
H<I>=P<I>xP<I>(<I>,<I>)L*1[condition]        for 2D L*1 histograms.
```

Note that the bits-per-channel specification may be omitted, in which case the defaults are I*4 for 1D- and I*2 for 2D-histograms.

Valid examples of histogram definitions are:

```
STRING                          COMMENT

H1=P1(7)I*4[]                   !1-D hist. of parameter 1 with 128 channels
H11=P3XP9(7,6)[C01]             !2-D hist. 128 by 64 channels gated by C01
H12=P1xP66(8,8)[C02+C03]        !2-D hist. gated by the AND of C02 and C03
H6=P10(12)[C01+(C02.C08)]       !1-D hist. of parameter 10 with 4k channels
H42=P8xP2(8,7)L*1[]             !2-D hist. (byte)
```

The comments should not be included when the strings are input, but are written here to explain the string meaning. Invalid examples followed by the reason are:

```
H1= P1(7)              !does not have the condition string,
H02=P01XP04(5)[]       !incorrect format in the dimension string,
H3=P6XP04(5)I*3[]      !illegal bits-per-channel specification.
```

Raw parameter numbers are always scaled to the histogram length. Thus any histogram should contain the full range of the parameter. No "gain" or "offset" is provided for, but a pseudo parameter could be written to give this facility if desired.

Within the limits of the AED core space (usually 1 Mbyte), one may define up to NHMAX histograms, and have a maximum of NCMAX different conditions, where NHMAX and NCMAX are defined in [SARA]DEFCOMMON.FOR.

A given condition string can contain up to 8 conditions, which are denoted by the charcter "c" followed immediately by the condition number in I2 format. To use the negation of a condition use "n" in place of the "c". Operators are denoted by single symbols and are listed below in their order of precedence:

symbol	operator	meaning
-	NAND	NOT a AND NOT b
+	AND	
/	NOR	NOT a OR NOT b
~	EOR	(a AND NOT b) OR (NOT a AND b)
.	OR	

The implied NOT operator ("n02" meaning NOT c02) has the highest precedence. To force a particular order of execution, use brackets, "(" and ")", where necessary. For example,

```
[n01+(c02.c03)]
```

means a histogram would be incremented if condition 2 or condition 3 is true and condition 1 is false. Apart from within a condition, spaces are skipped over and illegal characters will be ignored. Parentheses must match. Thus:

```
[c01 . c03 + c04]   and   [c01.(c03+c04)]
```

are valid (and equivalent), whereas

```
[c 01.c03+c04]  and  [n02 + c08)]
```

are not valid.

III(f). Main Command Level.

SARA types "SARA>" to signify that it is ready to accept commands as described below. Not all commands available on a given version of SARA are described below. In general, commands are 4-letter character strings. Most parameters may be entered on the same line as the command; if the parameter is not entered, it will be prompted for. Thus

SARA> DHIS 7<cr>

is a shortened version of

SARA> DHIS<cr>
Histogram to delete? 7<cr>

Some "show" commands (except for SBUF and SCON) have hidden parameters which, if omitted, take default values of the minimum and maximum quantities, respectively. Thus,

SARA> SSUM<cr>

will cause the sums of all defined histograms to be displayed, whereas

SARA> SSUM 1 10<cr>

will cause the sums of any defined histograms in the range one through ten to be displayed.

The valid responses to SARA> are:

AHIS <spc. filename> <hist. #>
-- allows data written to disk with the WHIS command to be read back and added to an existing histogram. The histogram sizes must be compatible. "AHIS ALL" is (almost) the inverse command to "WHIS ALL" and is useful in certain situations (see XAED command).

AUTO <option #>
-- this is for setting up "auto commands" which are executed at the end of a playback run. See section IV(e) for details.

CCON <hist. #> <new condition string>
-- allows the conditions on a specified histogram to be changed.

CHIS
-- prompts for a histogram definition string (see section III(e)), and then for the histogram name.

CIVA <int. var. #> <new value>
-- similar to CVAR (q.v.) except this is for the integer*4 array of variables.

COMP <old hist. #> <new hist. #>
-- command to create a new histogram that is the same as a specified existing histogram except that it is compressed by a factor of two. This only works for 1-d histograms.

CVAR <real var. #> <new value>
-- changes the value of a real*4 variable. "ALL" will cause all real*4 variables to be changed to the same value. When SARA is exited, changes to variable values are lost unless a new definition file was written.

DBUF
-- disables the display of data buffers (inverse of SBUF).

DCON <condtn. #> <hist. #>
-- deletes a condition. You must also enter the histogram number on which the condition was set. If that histogram no longer exists, enter a "0".

DHIS <hist. #>
-- deletes a histogram. The AED common area is rearranged to fill the area vacated by the deleted histogram.

DISP <hist. #> [<ymax>] [/LOG]
-- gives a rough display of 1D histograms on the TTY. This is designed for checking the binning process when the AED is not available. DISP/LOG will give a logarithmic counts scale (default is linear).

HELP
-- types a list of valid SARA> commands and a brief description.

NDEF <defn. filename>
-- allows a restart with a new definition file to be read from disk. Use of this command will erase all histograms and restart with the new setup file.

RHIS <spc. filename>
-- allows a histogram previously written to disk via WHIS (q.v.) to be temporarily read back into the display area. The memory may be lost when other histograms are created or deleted.

SBUF
-- displays the first 60 words of all data buffers received by the histogramming program. An aid for debugging. You may optionally request the event data and the bitmask for specific events to be typed out, and/or for LAM timeouts to be reported.

SCON [/FULL]
-- will display the current condition definitions. "SCON FULL" will display the defining coordinates of each condition as well.

SHIS [<i1> [,<i2>]]
-- will display the current histogram definitions on the screen. The full command is "SHIS i1 i2" where i1 and i2 are the lower and upper histogram numbers to be displayed. If omitted, they default to 1 and the maximum respectively.

SIVA [<i1> [,<i2>]]
-- displays the current values of all the integer*4 variables.

SPAR [<i1> [,<i2>]]
-- will display the current parameter definitions on the screen. N.B. The pseudo descriptions are only for bookkeeping and have no

effect on the executable program. To check what is really in effect, look at the source file for DATAR.

SSCA [<i1> [,<i2>]]
-- will display the scaler sums. You are advised to ignore these while taking data, since the scaler stream may be sampled on-line.

SSUM [<i1> [,<i2>]]
-- displays the total number of counts in defined histograms. For convenience, the total number of counts excluding channel zero are also given.

STAT
-- displays the current run number and taping status. The sampling rate, which is calculated by taking the ratio of the number of buffers analysed to the sequence number of the last buffer received, is also displayed.

STOP
-- stops histogram binning and causes SARA to exit. If you wish merely to use the VAX for some simple operations, typing CTRL/Z is appropriate - see section III(g).

SVAR [<i1> [,<i2>]]
-- displays the current values of all the real*4 variables.

WDEF <defn. filename>
-- writes the current histogram and condition working set to a disk file. It is advisable to use this command to save desirable changes in the definitions, in case of a program crash (but see III(b)).

WHIS <hist. #,[hist. #,..]>
-- writes a spectrum (histogram) on disk using compressed format. The spectrum may then be plotted, for example. A response of "ALL" to the prompt 'Spectrum number to write on disk?' will dump all defined spectra to disk with filenames S<ihno>R<irun>.SPC . A list of spectra, separated by commas, may also be entered. The default extension "SPC" may be changed by use of an XOPT command.

WSCA
-- writes the VAX scalers to a disk file "SCALER.<irun>". The filename may be changed by use of an XOPT command.

WSPE <hist. #>
-- used to write a given spectrum to a disk file either with the format: channel no. (I13) no. of counts (I13) or as an unformatted file with 256 block records. The (2I13) format option is not recommended for 2D histograms, since the resulting files may be huge.

WSUM [<i1> [,<i2>]]
-- writes the total number of counts in defined histograms to a disk file "SUMHIST.<irun>" .

XAED
-- disconnects or reconnects the AED display area to SARA's histogramming memory. The AEDTSK will be stopped or restarted as appropriate. N.B. The current histogram memory will be erased, so it is best to use this command before beginning a new run. However, "WHIS ALL" may be used to dump all histograms on disk; "AHIS ALL" can then be used after "XAED" to restore them.

XOPT <option #>
-- puts you in a second command level to access some rarely-used commands, such as setting the run number manually, changing default file extensions or showing the AED memory usage.

ZERO <hist. # [,hist. #,..]>
-- zeroes specified histograms. "ALL" will zero all histograms. A list separated by commas, e.g. "ZERO 1,3,5,7", is also valid.

ZSCA
-- will zero all the scalers. This only effects the scaler sums in the VAX and will have no effect on the LSI scalers which are zeroed when the run is changed.

III(g). Accessing DCL from SARA.

This may be effected by typing CTRL/Z whilst in SARA> mode. Histogram binning is not interrupted if data taking is active (but tape reading WILL be interrupted). DCL commands such as EDIT and PRINT are thus readily available.

A monitor-level command PRTHIST will run a routine that plots a spectrum held in the AED memory. Note that PRTHIST is not a valid response to the SARA> prompt - you must CTRL/Z first. The plotting routine accepts both one- and two-dimensional spectra. Unless /NETWORK is specified, the plot is queued to LPA0: (and printed some time later).

To reconnect SARA to the TTY, type

SARA

... and you should be back in SARA> mode. In playback mode, DCL is reached by spawning (and subsequently killing) another subprocess. In this case a CTRL/Y will stop SARA and LOG at any time.

III(h). Format of files written to disk via "WHIS".

One dimensional files are written in MUSORT format. Two dimensional files are written in a compressed format with channels containing zero counts suppressed. Details on how to read the two dimensional spectra into a FORTRAN program may be found in [HANCHAR.SPCLIB]ANINTRO.DOC and FORMAT.DOC (use SPCREAD or SPCREAD_ADD

from the library on that account). An example program that reads 1-d I*4 spectra is given in Appendix 3.

IV. PLAYBACK OF TAPES USING THE PROGRAM LOG.

IV(a). The Tape-reading Program.

The program LOG reads records (or "buffers") from an event-by-event tape and sends them into a "mailbox" which may be accessed by another program such as SARA. LOG should not be confused with the ROUTER system of data acquisition routines used to take data on-line. However, the same histogramming program SARA is called up in both cases.

When reanalyzing tape data, additional or different pseudo parameters may be desired; in that case DATAR should be edited and SARA relinked. (Of course, the number of non-pseudo parameters must be the same as written on-line). The definition file should be amended to take into account any changes in DATAR.

IV(b). Mounting Tapes for LOG.

Before running LOG, you should put a tape on the drive and issue a mount command, eg.

```
MOUNT/FOREIGN MFA
```

(if you are using drive MFA1: on the analysis machine, you must also type

```
ASSIGN MFA1: MFA0:
```

after mounting the tape.) To start the analysis program type:

```
RUN [SARA]LOG
```

You should answer "YES" to the question "Will you read data from tape?". In answer to the prompt "What is the name of the sub-process to start?", type SARA (this will call up SARA.EXE from your directory).

IV(c). TAPE mode commands.

When replaying tapes, SARA operates in two modes. A data mode for histogram binning which is the same as the mode described in section III and a tape control mode for tape commands. To denote the mode the program prompts

```
SARA> -- to signify main option mode, or
TAPE> -- to signify tape control mode.
```

The following important SARA> mode commands are available in off-line SARA:

TAPE -- causes the program to move to tape mode. Tape reading will be stopped if active.

TAPS -- stops tape reading. The mode is left as SARA>.

The valid TAPE> control commands are:

AREC -- Will advance an input number of buffers on the tape.

ARUN -- Will advance or rewind an input number of runs. Use negative numbers to rewind.

BRUN -- Will move the tape back an input number of runs. Negative numbers will move the tape forward the given number of runs. "0" moves to the start of the current run. If you BRUN to the start of the tape you will get an error message. REWI should be used to go to the start of the tape.

DATA -- causes the program to move to data mode

HELP -- type a list and brief description of TAPE> commands.

MREAD -- initiates a sequence of reading multiple runs from a tape without stopping at each EOF. Prompts for the total number of runs in the sequence and which ones to read. If the first run on a tape is included, it should be entered first; any order is acceptable for the others. The user can pause the playback at any time with the TAPS command, but when tape reading is resumed, the MREAD sequence will still be in effect. To abort it, use SREAD.

NRUN -- positions the tape at the beginning of a specified run. If the run number isn't found, an appropriate message is issued.

READ -- Begins reading data from tape, starting at the current tape location. Playback is stopped when the end of run is found. This command automatically sets you back to SARA> mode.

REWI -- rewinds the tape.

RREC -- Will read one buffer from the tape.

RSBF -- Read data buffers up to the next scaler buffer encountered.

SREAD -- Aborts the MREAD sequence. However, playback of the current run is continued up to the EOF.

UNLO -- Rewinds and unloads the tape.

IV(d) Running SARA without the AED display task.

There may be occasions when it is not necessary to display histograms. For instance, once all the gates are determined, a long playback run might be performed with no need to check the appearance

of spectra until the run is ended (at which point the histograms would be written to disk). In such cases SARA could be started without the AED (or the AED could be turned off via the XAED command after replaying a portion of the run), leaving the AED free for other users. To check that histograms are being incremented, the SSUM and DISP commands may be useful.

Note that SARA may also be used as a stand-alone program to display spectra. Simply type

```
RUN [SARA]SARA
```

and use the RHIS or AHIS commands to read back spectra from disk.

IV(e). Use of AUTO commands.

AUTO commands are instructions to be performed at the end of a tape playback run. The main purpose is to relieve users from having to wait for the end of a playback to make sure histograms are written to disk or scalers recorded. They could also be useful in ensuring that a particular sequence of instructions are executed for each playback (spectra and scalers zeroed, etc.).

To create AUTO commands, first type AUTO when in SARA> mode (this can be done at any time). From the menu, select 1. You will be prompted for commands line-by-line, e.g.

```
[1]: WHIS ALL
[2]: ZERO ALL
[3]: <CTRL/Z>
```

Note that <CTRL/Z> terminates the input of command lines (the limit is 20 lines).

The following SARA commands presently work in AUTO mode:

NDEF STOP SSUM STAT WHIS WSCA WSUM ZERO ZSCA

In addition, the commands SKIP and RESU(ME) are valid AUTO commands (see below).

Any valid commands will be executed whenever the end of a playback run is reached. You can also force the commands to be executed "manually" at selecting 2 in the AUTO menu; this may be done at any time. To list the present AUTO commands, select 3 in the menu. To cancel (erase) all AUTO commands, select 1 and type <CTRL/Z> with no commands.

The special commands SKIP and RESU(ME) are intended for control of which AUTO commands are executed during an MREAD sequence. If SKIP is found in the series of commands, the subsequent commands will be ignored (skipped over) if, and only if, the program is in the middle of an MREAD tape sequence. If an MREAD sequence is not active (READ mode playback) or the last run in the MREAD sequence has just finished, all the AUTO commands will be executed. If a RESUME command is found after a SKIP, the commands after the RESUME line will be executed regardless of the status of MREAD.

The other AUTO commands are similar to equivalent SARA> mode commands, with some differences as noted below.

NDEF -- must be followed by a line giving the name of the definition file, e.g.

```
[1]: NDEF  <-- command
[2]: ASETUP.DEF  <-- file name
```

Since one does not usually want the name definition file read in over and over, the above example would not be very useful. However, if you have definition files on disk with names containing the appropriate run number, and you are reading runs in an MREAD sequence, you may specify the filename with the run number replaced by <MR+1> (the brackets "<>" are necessary). When the AUTO commands are executed, this will be translated into the number of the next file in the MREAD sequence.

ZERO -- The following formats are available:

```
      ZERO ALL
      ZERO 1,2,4,12 ...
but not (at present)
       ZERO
       1
```

STOP -- will stop SARA (LOG will also terminate if active)

WHIS -- works much as in interactive mode, e.g.: to write all histograms with S%%%R%%%.SPC

```
      WHIS ALL
```

to write a series of histograms with automatic file labeling and titling

```
      WHIS 1,3,52
      <cr>
      <cr>
otherwise
      WHIS 7
      SPECTRUM.DAT
      TITLE LINE
```

will work, but give the same name for each run.

STAT, SSUM, WSUM, ZSCA, and WSCA -- as in interactive mode.

V. NOTES

APPENDIX 1 : Example Definition File for SARA

```
 PARAMETERS
          24                                  !NSCALER
 1,'      PC1L','ADC  ',12, 1   !PNUM,PNAME,PTYPE,PNBITS,PEVENT_TYPE
 2,'      PC1R','ADC  ',12, 1
 3,'      PC2L','ADC  ',12, 1
 4,'      PC2R','ADC  ',12, 1
 5,'RF/S320TOF','ADC  ',12, 1
 6,'      IC1 ','ADC  ',12, 1
 1,'      IC2 ','ADC  ',12, 1
 8,'ACTCOL TDC','TDC  ',12, 1
 9,'   MONIT 1','ADC  ',12, 2
 10,'   POS 1 ','PSEUDO',12, 1  !PNUM,PNAME,PTYPE,PNBITS,PEVENT_TYPE
     P10=P1/(P1+P2)              !DESCRIPTION OF PSEUDO PARAMETER
 11,'   POS 2 ','PSEUDO',12, 1
     P11=P3/(P3+P4)
 12,'  ICTOTAL','PSEUDO',12, 1
     P12=(P06+IV(4)*P07/IV(5))/2
 13,'  ETPOS1 ','PSEUDO',12, 1
     P13=( P1 + P2 )/2
 14,'  ANGLE  ','PSEUDO',12, 1
     P14={IV(12)-(P10-V(1)*P11)}*2**IV(13)
 -1
 VARIABLES
 REAL
      1, 0.81400 , ANGLE MATCH
 -1
 INTEGER
      4,         4, DE GAIN MATCH
      5,         4, DE GAIN MATCH
      12,      400, ANGLE OFFSET
      13,        5, 2**N FOR ANGLE
 -1
 HISTOGRAMS
 H1=P10(12)[]
  POS 1
 H2=P9(9)[]
  MONIT 1
 H3=P12XP5(8,8)L*1[]
  RF/S320TOF VS. ICTOTAL
 H6=P10XP14(7,5)[]
  ANGLE VS. POS 1
 H20=P10(10)[C01] C14 TOF/DE GATE
  POS 1
 -1
 CONDITIONS
 C 1=P12XP 5[NTYPE= 4,NPOINTS=  4](NHIST= 3)
   896  480
  3264  480
  3296 1536
   896 1504
 c01 contour
 -1
```

APPENDIX 2 : THE AED COMMANDS

a) ACCEPT - accepts the last contour, band, or gate that has been drawn, and send that set of points to the data acquisition process for additional processing.

b) BAND - draws on a two dimensional spectrum a set of line segments that define one side of a band. Each point that is selected is marked with an X and a straight line is drawn to the preceding point. A point in error can be erased by the 'e' command. The whole group of points can be rejected by the 'z' command. The band is not closed.

c) CONTOUR - is to be used on a two dimensional spectrum to enclose an area of interest. The method is to select a group of points one at a time in order to enclose an area. Each point that is selected is marked with an X and a straight line is drawn to the preceding point. A point in error can be erased by the 'e' command. The whole group of points can be rejected by the 'z' command.

d) DISPLAY - displays a spectrum in a selected window. The method is to move the cursor to the desired window, type 'd', and then enter the spectrum number.

e) ERASE - removes the last point generated by the contour, band, or gate commands.

f) FULL - temporarily displays the selected window on the full screen.

g) GATE - sets points on a one dimensional spectrum. A pair of points is used as gate limits (inclusive) for peak selection.

h) HELP - not implemented at present.

i) INTEGRATE - sums the number of counts in the selected contour for two dimensional data, or the number of counts in the selected gate for one dimensional spectrum. The counts near the cursor are also displayed.

j) - write the current AED histogram (window) set-up to a save file

k) - read back in the AED histogram (window) set-up from a saved file.

l) LINE - selects one point on a two dimensional spectrum through which a horizontal line is drawn. A point in error can be erased by the 'e' command. The line of points can be rejected by the 'z' command.

m) MODE - defines the display mode for that window. See [AU]AEDTSK.DOC for details. Mode 8 gives a logarithmic scale, for example.

n) NAME - enters the name of the current gate, band, contour. The field is 20 characters long and is transmitted as part of the data with the 'a' command.

o) CIRCLE - draws a circular contour on a two dimensional spectrum. The method is to first select the center by typing 'o' once. Next, move a distance away and type 'o' again to define the radius. The circle can be rejected by the 'z' command.

p) POSITION - shows the cursor position in channel number and the number of counts in that channel.

q) QUERY - gets text information about a spectrum.

r) REFRESH - selects a refresh time period for a selected window. This is done by moving the cursor to the desired window and typing 'r' and then entering the time period.

t) TYPE - enters commands longer than one character. Any commands not understood by AEDTSK are transmitted to the data acquisition process. At the current time, all AEDTSK commands are one character.

u) UPDATE - provides a one time refresh of a display window. This is done by setting the cursor on the desired window and typing 'u'.

v) VIEW - rolls the display so the text area is visible. The command acts as a flip-flop, i.e. the second 'v' returns the display to normal state.

w) WINDOW - selects the screen format. The display is rolled to the text area to permit the entry of the division factor for each axis of the display.

x) EXPAND - expands either a one dimensional spectrum between two selected x-axis points, or a two dimensional spectrum in the rectangle formed by the two points. The two points are considered to be opposite corners of the rectangle. To get back to the un-expanded view, use the DISPLAY command (or type 1,2...9 for a single digit spectrum).

y) CUT - selects an upper and lower cut level on the display.

z) CANCEL - kills the CONTOUR or CIRCLE command.

1,2..0) NUMBER - selects the spectrum to display. The method is to place the cursor in the window and type the number of the wanted display. Note that zero (0) is to remove a spectrum from the display.

?) used to display on the screen the type, sizes, and offsets to all known spectra.

/) displays on the screen the window information

-) decrements the display scale factor.

+) increments the display scale factor.

\) restarts the AEDTSK.

If the AED hangs (symptom is lack of cursor), try pressing RESET button once followed by the RETURN key. If this doesn't work, a suitable combination of RESET twice, "\" and RETURN will usually get AEDTSK restarted.

APPENDIX 3 : EXAMPLE OF A PROGRAM TO READ 1D SPECTRA

```
      SUBROUTINE SARAREAD(DATA,NPTS,HTITLE,IERR)
C--
C--   SUBROUTINE TO READ A DISK FILE IN THE FORM WRITTEN BY THE
C--   WHIS COMMAND IN SARA.EXE. THIS IS FOR A 1-D I*4 SPECTRUM
C--
C--   OUTPUT VARIABLES:
C--          DATA(1..NPTS) --  COUNTS IN CHANNELS
C--          NPTS          -- TOTAL NUMBER OF CHANNELS
C--          HTITLE        --  TITLE LINE OF SPECTRUM
C--          IERR          -- SET TO -1 IF SUCCESSFUL, 0 OTHERWISE
C--
      DIMENSION DATA(8193),HTITLE(20),EFILE(10)
      INTEGER*2 LOAD2(512)
      INTEGER*4 LOAD4(256)
      BYTE TITLE(80)
      EQUIVALENCE (LOAD2(1),LOAD4(1))
      EQUIVALENCE (TITLE(1),LOAD2(1))

      IERR = 0
      TYPE *,'ENTER FILENAME'
      ACCEPT 60,EFILE
60    FORMAT(10A4)
      OPEN (20,FILE=EFILE,TYPE='OLD',ACCESS='DIRECT',
     1   RECORDSIZE=256,READONLY)
      READ(20'1) LOAD2
C--   CHECK IT'S A 1-D FILE
      IF (LOAD2(72) .EQ. 1) GOTO 110
      TYPE *,'Spectrum is 2-D, data not read.'
      CLOSE(20)
      RETURN
110   ENCODE(80,120,HTITLE)TITLE
120   FORMAT(80A1)
      NPTS = LOAD2(74)
C--   LOAD THE SPECTRUM
      IRECNUM = (NPTS*2-1)/512 + 1
      DO 200 I=2,IRECNUM+1
      READ(20'I) LOAD4
      DO 200 J=1,256
200   DATA(J+256*(I-2)) = LOAD4(J)
      CLOSE(20)
      IERR = -1
      END
```