

LA100 Application Note 17: Reading & Saving Sequence Results

1. Introduction

Sequence testing involves the sending of a sequence of audio test signals for the complete and automatic evaluation of a system. The Lindos LA101 can generate the sequence tests and the LA102 can receive them and produce test results. The LA101 and LA102 are completely portable and self-contained - they do not need a computer to control the automatic test sequences and programming a test sequence into the LA101 is simple. Further details are given in chapters 3 and 4 of the LA100 manual. All page references for the LA100 manual refer to the fourth edition which is available from Lindos.

Although the LA100 can display and print the sequence test results itself, the results can also be transferred to a computer via the built-in serial interface. This allows high resolution graphs to be displayed, and allows results to be stored on disk. Lindos can supply Support Software for various systems:

Order Code	Computer	Current Version
DSK1	BBC Computer	V4.1*
DSK2	IBM PC compatible with MSDOS	V1.3*
DSK3	Acorn Archimedes with RISC OS	V6.0

*These versions do not use the new file format described here, but the next versions will.

Other machines may be supported later (in particular, the Apple Macintosh).

The LA100 remote commands described in this booklet are only available in LA100 V6.0 and later software. All software updates are available, free of charge, from Lindos Electronics.

1.1 Limitations in Earlier Software

The Lindos Support Software allows results to be stored on disk, printed, or checked against a pass/fail tolerance file. However, early versions of the Support Software each used their own file format which prevented the data from being transferred between computers. More seriously, it was impossible to export the results to other applications.

Furthermore, the remote commands to read sequence results (R? and S?) could not read results from 'multiple compartments' (see LA100 Manual, section 4.17, page 79). This meant that test sequences containing more than one frequency sweep or more than one distortion segment, for example, could not be read. The old remote commands, while useful for programmers writing their own test software, were not very useful when reading a set of results in their entirety as the computer software needed to know the details of each test segment.

All of these problems are being addressed in the latest Lindos software:

1.2 New Results File Format

A new universal file format has been defined to enable results files to be transferred between different computers, and to facilitate the exporting of results into other applications. The new file format contains more information, yet is more compact, allowing more results to be held on a single floppy disk. Being transferable between machines means that results saved on a PC can now be loaded on an Acorn Archimedes, and vice versa.

The DSK3 RISC OS Support Software (V6.0) already uses this new format and the next version of the DSK2 PC Support Software will also use it. (The DSK1 Support Software will only be updated if there is sufficient demand). All other software which reads results from the LA102 should save in this format and all software to process results should use it.

The new Lindos Support Software can convert old results files into the new format, so existing results files may still be used.

1.3 Exporting Results into Other Applications

Adopting the new file format means that it is now very easy to produce text or CSV (Comma Separated Value) files which can be exported into other applications. The new Support Software can generate text and CSV files, and the LA100 can return results in either format.

Modern wordprocessors, spreadsheets and DTP (Desktop Publishing) systems are becoming very powerful and most can import CSV, text and graphics files. Being able to load the sequence results into a spreadsheet enables very sophisticated processing, while loading graphs into wordprocessors and DTP systems allows test results to be embedded in reports, as well as allowing a great deal of control over the final layout.

1.4 LA100 Software

The new LA100 V6.0 software provides a remote command to read all of the textual results in a single operation, making it very easy to generate the new results files. Furthermore, the remote software no longer needs to know the details of individual test segments and will work with any new segments which are introduced into future LA100 software.

The old LA102 2R? and 2RR? commands (LA100 manual, page 145) have been retained for backwards compatibility, but should not be used by new software.

2. LA102 Remote Commands

The new command 2SR? command (LA100 manual, page 145) reads the sequence results as a single block of ASCII text terminated with a CTRL-Z character (ASCII code 26). By default, every line is terminated with a Line Feed character (ASCII 10), and no other control characters are included. The format resembles the sequence results printout which the LA102 can produce and is intended to be both machine and human readable.

<pre> LINDOS AUDIO SEQUENCE TEST SOURCE 0518E LA102 0518 V6.0 SEGMENTS +TDZ TD TEST LEVEL OUT [T] 0dB +0.01 -6.65dBu DISTORTION (2f-22k) [D] +8dB 100Hz -87.9 -89.0 1kHz -89.5 -91.4 6.3kHz -84.4 -88.3 PHASE (mean) [Z] 0dB 40Hz -4d 100Hz -5d 1kHz -11d 6.3kHz -42d 10kHz -58d </pre>	<pre> LINDOS AUDIO SEQUENCE TEST SOURCE 0518E LA102 0518 V6.0 SEGMENTS +TDZ TD TEST LEVEL OUT [T] 0dB , +0.01 , -6.65dBu DISTORTION (2f-22k) [D] +8dB 100Hz , -87.9 , -89.0 1kHz , -89.5 , -91.4 6.3kHz , -84.4 , -88.3 PHASE (mean) [Z] 0dB 40Hz , -4d 100Hz , -5d 1kHz , -11d 6.3kHz , -42d 10kHz , -58d </pre>
--	---

Fig. 1 Results text read by SR?3 (left) and same results text in CSV form read by SR?7 (right)

Bit	Value	Action
0	1	If this bit is set then page breaks and margins are suppressed making the text suitable for processing and storage on disk. If this bit is clear the text is formatted as though it was being sent to a printer, leading spaces are included at the start of every line and the text is divided into formatted pages using configuration N to determine the page length and configuration M for the top margin. Formatting would normally only be used when passing the text directly to a printer.
1	2	If this bit is set then embedded <i>graph handles</i> are included in the text (see ‘Graph Results’ below).
2	4	If set the data fields are separated with a comma to produce a CSV file (see CSV Format below)
3	8	If set terminate text lines with CR and LF (the PC/MSDOS convention). If clear terminate text lines with LF (the Unix and RISC OS convention).
4	16	If set force the distortion and level results to be given in dB (as though LA102 configurations D2 and U0 were set), otherwise use configurations D and U to determine the units. The former would be used when the computer software knows about converting from dB to percent, and when tolerance testing is going to be applied to the results. Note that this bit has been inverted since the first edition of APP17 (no software was released using the old convention).
5	32	If set suppress tolerance testing, and don’t include any tolerance information.
6-7		Not allocated (must be zero).
8-15		Include printer control codes and graphics bit map data for frequency response graphs. These bits represent the printer type, as for configuration P, but will normally be zero. Some care is needed as the control codes may include an ASCII 26 byte (either in an escape sequence, or as part of the bit map graphics data) before the true end of file character, so the only way to be sure that the end of file has been reached is to timeout if no data is sent for 2 seconds. If control codes are included the resulting data cannot be processed easily and the only real use is if it is being send directly to a printer.

Fig. 2 Bit Allocation for Option Number n (in Remote Command SR?n)

An extended form of this command, 2SR?n allows an option number to be specified to control the precise format of the returned data. Each bit in the option number controls a particular feature and by default all bits are 0. The bits are described in fig. 2.

The command SR?3 would normally be used when reading the results text for storage in a file. This will suppress the margins and pagination and include the embedded graph handles. SR?11 might be used on a PC to make the LA102 terminate the lines with CR and LF (the MSDOS standard). For example, if the LA102 has just received the sequence TDZ, an SR?3 or an SR?7 command will return results similar to those in fig. 1.

3. Graph Results

Although the results text includes frequency response results, only a few frequencies are included because the full list would be very long. These frequencies are the same as for normal printouts and Table 3.2 on page 39 of the LA100 manual lists the frequencies for each frequency response segment. Attempting to use these few values to plot a frequency response graph would produce very crude results.

The full 256 samples of frequency response results can be read from the LA102 using the S? command, as described on page 146 of the LA100 manual. The results are returned in binary to keep the size down. Each sample is returned as 2 bytes, so two sets of frequency response results require about 1Kbytes of data, taking 1s to read at 9600 baud. (By comparison, if the same results were returned as ASCII text they would take at least 7 bytes per sample).

When reading textual results using the SR? command, described above, it is possible to include embedded graph handles in the results text by setting bit 1 in the option number. This adds a ^ symbol and a graph handle, for each channel, to the end of each frequency response segment title. For example:

```
FREQUENCY RESPONSE [X] 0dB^1^2
30Hz -1.11 -1.11
40Hz +0.55 +0.55
50Hz +0.55 +0.55
```

In this example, there are frequency response results for two channels, so two handles are given. The handle number should be quoted in the S?n command, where n now represents the graph handle. Extending the S? command in this way allows multiple sets of frequency response results to be read from an LA102 (note that the results may contain up to 10 graphs if configuration C5 has been set in the LA102 - see section 4.17, page 79 of the LA100 manual).

4. Procedure for Reading Sequence Results

To read all of the results from an LA102 a program should perform the following steps:

1. Put the LA102 into remote mode (section 9.4, page 124, LA100 manual).
2. Send an ID? command to read the identity of the LA102.
3. Read the identity into a string (terminated by a CR character).
4. Ensure that the LA102 has V6.0 or later software. Do this by scanning the identity string until a capital 'V' is found, and read the following digits (one or two digits will precede the decimal point). If this number is less than 6, the LA102 does not support this format and the results cannot be read, so generate an error and go to step 9.

5. Send an SR?n command (n is usually 51 or 59, but could be 1 or 9 when reading simple results).
6. Read characters into text buffer up to and including the terminating ASCII 26 character.
7. Scan the text buffer for ^ characters. For each one that is found, read the following graph handle (note that the graph handle may be over 9, so all subsequent digits should be read up to the next non-numeric character).
8. Read the graph results (if any) using the S?n command where n is the graph handle. Repeat for each graph handle that was found. Note that the S?n command returns three lines of text, representing the start and finish frequencies and the number of samples. This is followed by 2 bytes per sample for every sample (this is always 256 in current software). This binary data should be held in a buffer, as it is, without processing.
9. Send a KB1 command to return the LA100 to manual mode.

If results files are to be created on disk it is important that the data returned from the LA102 is not processed in any way when it is read in (ie do not strip out the ^ symbol or the graph handles and do not convert the graph data to decimal). The results files use the same compact format, so it makes sense to keep the data in that form and only convert it when it is needed.

If results files are to be produced, it is recommended that a 32Kbyte block of memory is reserved to hold the results file and that the data from the LA102 is written into this block using the results file format described below. Saving and loading are then very simple - the results block is copied directly to or from disk. When displaying, processing or printing the results the same data block should be used.

5. Results File Format

The results file format described here is used by all Lindos Support Software from PC Support V2.0 and RISC OS Support V6.0 onwards. Support Software for other machines will also use the same format.

There are two types of results files: simple and complete. The simple files contain just the textual part of the results with no graph data and no embedded graph handles. They are completely human readable, containing only printable ASCII characters and may be loaded directly into a wordprocessor. The complete files contain the same textual part of the results, but they also contain embedded graph handles (where relevant) and the binary graph data. Lindos Support Software uses the complete file format, but can also export the simple form so that the results can be loaded into a wordprocessor.

5.1 Simple Results Files

The simple files are created by reading the results from the LA102 using an SR?1 or SR?9 command and copying the returned text into a file. This corresponds to steps 1-6 and 9 in the above procedure, but the end of text (ASCII 26) character is not included. The lines are terminated with an LF (if SR?1 was used) or CR LF (if SR?9 was used). Either may be used, and the choice depends on the usual format for text files on the system being used. This should make it easy to load the files into a wordprocessor on the same system. Results files are still compatible between different systems since all software which reads them should follow the guidelines below and allow for any combination of line terminators.

5.2 Complete Results Files

The complete files start with the same textual results as the simple files, with embedded graph handles, and ending with an end of text (ASCII 26) character. The rest of the file consists of a variable number of data packets. Each data packet starts with the following two lines which form a packet header. These lines are ALWAYS terminated with a CR (ASCII 13) character only.

type, format *type* is a string specifying the packet type, while *format* is a number.
n,m Size of packet data: *n* lines of text followed by *m* bytes of binary data.

The packet header is followed by *n* lines of text, each terminated with an LF character. This is followed by *m* bytes of binary data. Either one or both of *n* and *m* may be zero. See section 6.8 for dealing with unknown packets encountered in a results file.

5.3 The Graph Packet

The only packet currently defined is the 'Graph' type which has the following format:

<i>type,format</i>	Packet type (a case-insensitive string) and format number
<i>n,m</i>	Size of packet data: <i>n</i> lines of text followed by <i>m</i> bytes of binary data.
<i>xunits,yunits</i>	Two strings giving the units for the x and y axes respectively.
<i>handle</i>	Graph handle number
<i>f1</i>	Initial x value
<i>f2</i>	Final x value
<i>s</i>	Number of samples
< <i>m</i> >	bytes of binary data where <i>m</i> is always 2 <i>s</i> .

The following *format* numbers have been allocated:

0	Linear X and Y axes
1	Logarithmic X, linear Y axes
2	Linear X, logarithmic Y axes
3	Logarithmic X and Y axes

Only format 1 is currently used. Software which can handle this packet should plot a graph using the data in the packet. The x and y axes should be labelled in the units specified in the packet header (usually Hz and dB, but possibly Hz and degrees in future software). If the format is not 0-3 then the whole packet must be ignored by current software.

For example, a typical graph packet will look like this:

Graph, 1	Type is 'Graph', format is 1
5, 512	5 lines of text and 512 bytes of data in the packet
Hz, dB	Units are Hz on x axis and dB on y axis
2	Graph handle (will correspond to ^2 in textual results)
20	Start frequency in Hz
20000	Finish frequency in Hz
256	Number of samples (always 2 bytes per sample)

This is followed by 256 samples in the same format as the LA102 returns following a remote S? command. Each sample is 2 bytes, integer byte first, followed by fractional byte, using two's complement. See page 146 of LA100 manual for more details.

5.4 Generating Graph Packets

A graph packet should be generated for each set of the frequency response results. The graph packet is normally generated as the frequency results are read from the LA102 (see step 8 of section 4, above).

To produce the packet header, simply write the following text onto the end of the results file:

Graph, 1	Type is 'Graph', format is 1 (log x axis, linear y axis)
5, bytes	Always 5 lines of text, where <i>bytes</i> is $2n$ and n is the number of samples (as returned from the LA102).
Hz, dB	Units are Hz on x axis and dB on y axis
handle	Graph handle (as given by n in results text, and as used in S? n command).

This should be followed by the three lines of text and the binary data returned by the LA102 after an S? command. See page 146 of the LA100 manual for full details. Packet data lines must always be terminated with a single CR character (ASCII 13).

Note: The 3 lines of text returned by the LA102 must be read before the packet can be generated, since the number of samples n is needed in order to calculate the bytes value giving the number of binary data bytes in the packet. The actual binary samples may be copied directly onto the end of the results file as they are read from the LA102.

6. Reading and Processing Results Files

It is important that software follows these guidelines to ensure that it will cope with future developments and also that it will handle results files created by other systems.

6.1 File Buffer Size

A file buffer of 4Kbytes is sufficient to hold simple results files, while a 32Kbyte buffer is recommended for complete results files. Note that most results files will be less than 2Kbytes, but a very long sequence containing 10 frequency sweeps will occupy over 6Kbytes. Future LA102 or Support software may produce larger results files.

6.2 Line Terminators

The textual part of results files consists of lines of ASCII text, with each line terminated by a CR, LF, CR LF pair or LF CR pair. When a CR or LF character is found, an end of line has occurred. If the character was a CR character and the next character is an LF then the LF should be skipped. Similarly, if the character was an LF and the next one is a CR then the CR should be skipped.

Some care is needed here: Do NOT regard CR and LF as both meaning end of line, since a file using CR LF pairs as line terminators will then be regarded as having a lot of blank lines in between every real line! Similarly, do not automatically filter out all subsequent CR or LF characters once one has been found as that will have the effect of removing all blank lines!

Note that lines in data packets are always terminated with CR (ASCII 13) and you must not use the same routine for processing those lines, as it will fail if the first byte of binary data is 10!

6.3 File Header Lines

The first few lines of the results file, up to the first blank line, contain file header information. Subsequent lines represent the body of the results.

The first line of a results file always starts with the following text:

```
LINDOS AUDIO SEQUENCE or LINDOS AUDIO PROCEDURE
```

If neither are present then it is reasonable to reject the file. The rest of this line can contain a 40 character heading starting at character position 32 (ie spaces are used to pad up to this position). It will be filled in by the LA102 if the user has defined a heading in the LA102.

The second line gives information about the sending and receiving systems. The word 'SOURCE' is followed by the source ID (see section 4.3, page 67, LA100 Manual), as received from the LA101 in the FSK + segment. This is followed, at character position 32, by the text 'LA102' and the measuring set ID (the serial number and internal software version number):

```
SOURCE 0001Z           LA102 0518 V6.0
```

The third line contains the word 'SEGMENTS', one or more spaces and a list of segments received on each channel (the lists are separated by one or more spaces):

```
SEGMENTS +TX TX
```

Current LA102 software only returns three header lines, the fourth line is always blank, but it is quite permissible to insert extra non-blank lines here containing comments about the results. It is recommended that the comments be restricted to 4 lines, but there is no limit. A blank line should always follow the header lines, and must be totally empty. A line containing a single space is not regarded as being blank (this allows multiple comments to be separated by a line containing a single space). Tab characters are not used at all.

6.4 Test Results

The body of the test results follows with a few lines of text corresponding to each of the test segments that were received. Any line containing square brackets is a segment (or procedure) header line. Subsequent lines up to the next header line contain results for that test segment.

If the header line contains a single letter within the square brackets, the letter is the segment letter (as used by the LA100). If the brackets contain more than a single character, then this text represents the name of a computer controlled test procedure, and the results for that procedure follow. See section 9 (below) for more details of test procedures.

For example:

TEST LEVEL OUT [T] 0dB	Segment header line
+0.01 -6.65dBu	Test level results
DISTORTION (2f-22k) [D] +8dB	Segment header line, for segment D
100Hz -87.9 -89.0	Distortion segment results
1kHz -89.5 -91.4	
6.3kHz -84.4 -88.3	
PHASE (mean) [Z] 0dB	Segment header line for segment Z
40Hz -4d	Phase segment results
100Hz -5d	
1kHz -11d	
6.3kHz -42d	
10kHz -58d	
15kHz -73d	

6.6 Removing Graph Handles

It is recommended that the graph handles are removed from the text before printing or exporting as a simple results file. This can be achieved by scanning the test results (after the header) and removing all characters after a ^ to the end of line.

6.7 Results Columns

The precise format of the results lines is:

- 8 character measurement title
- 7 characters containing the results value for the L channel
- 2 spaces
- 7 characters containing the results value for the R channel
- Any other text

This is shown in fig. 2. Note that the results are right justified in the 7 character field width, but the decimal points may not be aligned, since some measurement have more decimal places than others: test level and frequency response results have 2 decimal places; distortion, crosstalk and noise have 1; phase results have no decimal places; W&F results have a variable number). The results value may also have a units suffix, such as d for degrees, or % for percent.

	1	2	3
Column:	123456789012345678901234567890		
Contents:	TTTTTTTTLLLLLLL	RRRRRRR	
Examples:	12.5kHz +0.01	-4.40	
	PK UNW -104.5	-104.6	
Key: T=Title, L=Left results, R=Right results.			

Fig. 2 The columns used in results files

6.8 Interpreting Results Values

Results values are always in ASCII decimal, and most, but not all will contain a decimal point. Some will have a units suffix of 'dBU' for decibels relative to 0.775V, 'd' for degrees or '%' for percent. Where no suffix is given, dB should be assumed.

The graph data is always in absolute units (dBU for frequency response graphs) and never normalised. Normalisation may be performed when the graph is drawn, but the results themselves should not be modified.

Distortion results should always be stored in dB in the results file, never in %. Distortion measurements have traditionally been quoted as a percentage expressing the level of the distortion products relative to the level of the fundamental. More recently there has been a trend to express distortion results in dB. The LA100 and all Lindos Support Software provide a configuration option to control whether distortion measurements are given in % (configuration D1) or dB (configuration D2). A procedure is given below (section 7.3) for converting the results to %.

6.9 End of Simple Text

Simple results files have no terminating character. If an end of file is reached reading a results file, then it is a simple results file and contains no data packets.

Complete results files use an ASCII 26 character to mark the end of text. This is followed by results data packets.

6.10 Data Packets

Results data packets are used to hold extra data which does not form part of the normal textual results, but which may be needed for other purposes, such as graph plotting. The format is described in section 5, above. Each packet has a packet header which specifies the type and format of the packet.

Software which does not recognize a packet type or format in a results file must not attempt to process the packet, nor must it generate an error. Unknown packets must be preserved, unchanged, to allow for future expansion. If the results are saved the unknown packet must be written back to a file unaltered.

7. Procedure for Processing Results for Output

When results are being displayed, printed or exported, it is quite permissible to output the textual part of the results file directly, without any modification. However, it is recommended that the following procedure is used to enhance the final output, wherever possible.

7.1 Tolerance Testing

A tolerance file contains a list of pass/fail limits for one or more of the sequence results values, with one tolerance string being applied to each measurement. The tolerance strings are normally shown in an extra column to the right of the sequence results and values which fail are marked with a '*'. The keyword 'TOLERANCE' is shown on the third line of the result, followed by the optional number, the name and a PASSED or FAILED message (see fig. 5.4 on page 87 of the LA100 manual).

The format of a tolerance definition is described in sections 5.3 to 5.5 (pages 89-92) of the LA100 Manual. A tolerance file containing the tolerance definition uses the same

format, but may use newline characters where spaces are permitted to make the file more readable. The rules given in section 6.2 about line terminators should be applied when processing tolerance files.

Tolerance files may be transferred to the LA102 to define a tolerance which the LA102 can use directly, or they may be used by computer software, such as the Lindos Support Software to allow sequence results to be checked against a tolerance in the computer. In the latter case the software should show the results in the same format as the LA102 (see fig. 5.4 on page 87 of the LA100 manual), and test each measured value against its corresponding tolerance. The printout format control described in section 5.5 of the LA100 manual should also be implemented. The procedure described below includes all of these features.

7.2 Bold Headings

To achieve the same appearance as a standard LA102 printout, some of the text should be shown in bold. The first line, all segment and procedure header lines (ie lines containing square brackets), and the keywords 'SOURCE', 'LA102' and 'SEGMENTS' in the header. A procedure for processing results in this way is described below.

When printing directly to a printer the bold effect is usually available as a control character or as an escape sequence. More sophisticated systems (eg Windows, or RISC OS) will use a printer driver program in the computer which will have its own way of producing bold text. In some systems it will be necessary to choose a bold version of the font.

Producing bold headings in text which is being exported to another application can be more difficult, although many applications can import files containing embedded font information. For example, Microsoft word can load RTF (Rich Text Format) files, while Impression can load DDF (Document Description Files). If there is any obvious standard on the system being used, it is desirable to use it, otherwise it should be left for the user to change manually.

7.3 Processing Details

This procedure allows the output to be modified as follows:

1. Bold headings can be used where available (note that this will not be possible when producing raw text or CSV files).
1. Distortion measurements can be converted to percentages where required.
2. A tolerance file can be applied and any failures marked with a '*'.

It is recommended that two text buffers are used - one containing the original textual results in dB, the other containing the output. It is important that this output file is never used to produce a Lindos results file, as it would not conform to the specification. Lines can be processed one at a time, by reading a line of text from the input buffer, modifying the line as necessary and then passing it to the output buffer. See section 6.2 for details of handling line terminators.

A third buffer should be used to hold the tolerance file if one is being applied.

The procedure is as follows:

1. Create and initialise these variables:

boolean <i>cvtpctflag</i> := FALSE	Flags % conversion on next line
string <i>linebuf</i> := ""	Contains line being processed
string <i>tolref</i> := ""	Name of test to look up in tolerance file
pointer to character <i>tolptr</i> := NULL	Pointer into tolerance buffer
string <i>tolstr</i> := ""	Tolerance string

Process file header lines:

2. Copy the header lines from the input buffer to the output buffer (up to and including the first blank line). The first line should be converted to bold (if possible), as should the keywords 'SOURCE', 'LA102' and 'SEGMENTS' – see section 6.3.
 3. Read a line into the line buffer *linebuf*.
 4. Test *linebuf* to see if it is a segment header line (containing a '[' symbol – see section 6.4).
- If not, go to step 10.

Process segment header line:

5. Set *tolref* to the contents of the square brackets. If this is more than one character then include the square brackets themselves (eg for [T] just use 'T', but for [PHASE] use '[PHASE]').
6. If a tolerance file is being applied then scan the tolerance file for *tolref* (ignoring text inside quotes). If found then set *tolptr* to point at the next character which is neither a letter nor a square bracket symbol, otherwise set *tolptr* to null.
7. Make the segment header line bold (if possible).
8. If distortion results are required in % then set *cvtpctflag* to TRUE if the segment header line contains the word 'DISTORTION' (case insensitive).
9. Go to step 16.

Scan tolerance file:

10. If *tolptr* is NULL go to step 14.
11. Read the next tolerance string, up to the next comma, letter, open square bracket ('[') or end of line (or file). Strip leading and trailing spaces.
12. If it is not empty then put this string in *tolstr* (otherwise do not change *tolstr* - this will cause the previous tolerance string to be used if none is given here).
13. If *tolstr* = "?" then set *tolstr* := "". This will suppress the '?' from the printout, and means that there is no tolerance for these results values.

Process results line:

14. If *tolstr*="" then go to step 17 to suppress this results line. If *tolstr* contains a tolerance string, check the results against it, and mark any which fail with a '*'. The results will need to be extracted from *linebuf* (see section 6.7). The '*' should be inserted after the results value. See section 5.3, page 89 of the LA100 manual for rules about tolerance strings.
15. If *cvtpctflag* is TRUE then convert the results values in *linebuf* to percent, as follows: Extract the results values (see section 6.7), convert them to % (see below) and write them back into *linebuf*. Take care to ensure that the same field width is used (see section 6.7).

Write to output buffer:

16. Write linebuf to the output buffer.
17. Repeat from step 2 until the end is reached (marked by an ASCII 26).

The percentage p corresponding to a decibel level d can be calculated from:

$$p = 100 \times 10^{d/20}$$

To convert from percent to dB, use:

$$d = 20 \log_{10} (p/100)$$

8. File Types and Extensions

Most operating systems allow an extension or description of a file. Sometimes this is a short 3 letter extension, as in MSDOS, whereas other systems may allow more characters.

The following identification should be used for the various Lindos files:

Data File	MSDOS extension	Unix extension	RISC OS type
Sequence results files	.RES	.res	SeqRes (&3F3)
Sequence definition files	.SEQ	.seq	SeqDef (&101)
Tolerance definition files	.TOL	.tol	ToIDef (&103)
Oscillator weightings	.WGT	.wgt	FrqWgt (&104)
Audio test procedure	.LAP	.LAp	LAProc (&105)

The RISC OS filetype numbers &101-&105 are provisional. The RISC OS DosMap command may be used to map DOS files onto the corresponding RISC OS type. For example:

```
*DosMap RES SeqRes
```

The sequence and tolerance definition files should contain plain ASCII text which can be easily edited in a text editor. If the host system does not allow easy editing of non-text files then these files should use the system text filetype. Note that most MSDOS editors allow files with any extension to be loaded (eg C source code normally has the extension .C), while most RISC OS editors will load a file which is double clicked with SHIFT held (double clicking on a sequence or tolerance file can then be used to execute the sequence or apply the tolerance).

9. Procedures - Computer Controlled Tests

Computer software may be written to perform other automatic audio measurements (procedures) which are not covered by the currently available test segments. Examples might include phase response, or long term monitoring of a particular parameter such as level or noise on a piece of equipment (eg a dropout test on a tape machine, or a noise test on a telephone line).

We hope to introduce a test procedure language in future Support Software which will simplify the control of an LA100 and allow test results files to be created. Meanwhile, software can be written in a standard programming language to control the LA100 (perhaps via a Lindos device driver), and produce results files which are compatible with the sequence results files described above.

Whether or not the Lindos procedure language is used, audio measurements made under computer control are referred to as test procedures. The results from test procedures should be stored in a results file, as described below:

9.1 File Format for Results from Test Procedures

The first line of the file must start with 'LINDOS AUDIO PROCEDURE' when the results were obtained automatically under computer control. The second header line should contain a description of the source and receiver, and the third line should contain some indication of what the test was. A blank line precedes the body of the results which should be in plain ASCII text.

The results may be divided into blocks, with each block preceded by a one line description containing the procedure name in square brackets. This is similar to the way sequences are divided into segments with each segment having a segment title and identification letter in square brackets.

Procedure names must be between 2 and 12 characters long. Single letter procedure names are not allowed since they would be confused with test segments. The format of the results should be the same as for segment results (see section 6.7).

A typical results file could look like this:

```
LINDOS AUDIO PROCEDURE TEST      SKIRRONICS DEVELOPMENTS
SOURCE Machine number 1          23 July 1993
Procedure test

GAIN CHECK [GAIN] 0dB
1kHz      0.00
PHASE RESPONSE [PHASE] 0dB
30Hz      +1.9°
40Hz      +3.1°
```

Where a large amount of data is collected the full set of results data should be stored in a data packet after the textual results (see section 5), and the text should contain a short summary. This technique is used for frequency response segments where the full list of 256 samples would take several pages. Only a few spot values are listed, but the full set of data is available in a data packet for graph plotting etc.

Note that it is quite permissible to use a data packet of the 'Graph' type to store results, and Lindos Support software will then automatically plot a graph of the results when they are loaded. For example, a long term level monitoring program might produce a graph header like this:

```
4,2000      4 lines of text and 2000 bytes of data in the packet
Graph, 0    Type is 'Graph', format is 0 (x and y both linear)
s, dB      Plot dB (on y axis) against s ( on x axis)
2          Graph handle
0         Start time in s
3600      Finish time in s (1000 samples in one hour, one sample every 3.6s)
1000     Number of samples (always 2 bytes per sample)
```

It is also permissible to add results for computer controlled tests into a standard Lindos results file, providing the resulting file conforms to the specification described above. This could allow a sequence test plus a phase sweep to be stored in a single results file, for example.

9.2 Tolerance Testing of Procedure Results

Procedure results look like segment results, but have a test name which is more than one letter long. To enable tolerance testing of procedure results, the format of tolerance definitions (see section 5.4, page 89 of the LA100 manual) has been extended. Wherever a segment letter is allowed in a tolerance definition it is now permissible to include a procedure name in square brackets.

For example, a tolerance file for the procedure results shown above could be:

```
"TAPE TOLERANCE"
 [PHASE] ±10, ±8, , ±4, , ±2, , , , , , , , , ±4, , ?
```

The algorithm described in section 7.3 (above) for processing sequence results will handle such tolerance files.

10. Further Details

10.1 Example Files

A floppy disc containing example results files is available from Lindos, upon request. Please specify the disk format you need.

10.2 Questions and Comments

This document describes many ambitious techniques and algorithms which have not yet been fully implemented by Lindos. The exact details may change as problems and areas for improvements are encountered, or in response to feedback.

If you have any comments, suggestions or questions, then please contact Lindos Electronics. We are always eager to help software developers, and users, and hope that the new file formats and test procedures will make the LA100 much more powerful and useful, as well as allowing it to be fully integrated into other computer systems.

Revision History

Second edition, 28th July 1993:

Section 2, SR?n option byte n has had bit 4 inverted, and bit 5 has been introduced

Third edition, 2nd September 1993

Section 8, RISC OS filetype &3F3 has been allocated by Acorn

Fouth Edition, 5th April 2007:

No content revision. Laid out and designed in new template format.

PMS 31.08.1993