

# SPR.FMT

PSIONICS FILE - SPR.FMT

=====

Format of Spreadsheet files

Last modified 1997-08-14

=====

This document describes the layout of SPR files for the SH3 spreadsheet.

A data file (also called a database file) begins with a 22 byte header of the following form:

Offset 0 (cstr): "SPREADSHEET"

Offset 16 (word): format version number

Offset 18 (word): offset value (meaning unknown)

Offset 20 (word): OPL runtime version number

The version numbers and offset value are all zero.

The rest of the file consists of records. All records have the form: Offset 0 (word): type of record

Offset 2 (word): size of data portion in bytes (L)

Offset 4 to L+3: data portion

Unless stated otherwise, records may appear in any order. Record types 11 and 12 should not appear in Series 3 spreadsheets - they were used by the MC. Types 13 onwards are new in the Series 3.

Unless stated otherwise, a cell reference is two words - column then row - and a range reference is four words - left, top, right, bottom, in that order. Each word is interpreted as follows. If the word is N, then:  $0 \leq N \leq \$1FFF$ : Nth row or column (so 0 is row 1 or column

A)

$N = \$8000$ : this row or column  $\$8000 < N < \$9FFF$ : row  $N - \$8000$  below or column  $N - \$8000$  to right

$\$E000 < N \leq \$FFFF$ : row  $\$10000 - N$  above or column  $\$10000 - N$  to left  
Values with the top bit set are only permitted in formulae. In some other circumstances a reference containing all \$FFFF values is used for "none".

Record type 1 holds formulae. A formula must come before any cell (record type 2) that accesses it, and the relative order of type 1 records should be preserved. The record takes the form:

Offset 0 (word): number of records using this formula  
Offset 2 (byte): length of formula (F)

Offset 3 to F+2: formula

A formula is stored using a Reverse Polish notation. In other words, to evaluate a formula, start with an empty stack, and then scan the formula in order; when finding an operand, push it on the stack, while when finding an operator, take its arguments off the stack, apply it, and push the answer on the stack.

Operators are represented by single bytes:

1 = <      5 = <>      9 = \*      13 = -      17 = &

2 = <=      6 = =      10 = /      14 = NOT

3 = >      7 = +      11 = \*\*      15 = AND

4 = >=      8 = -      12 = +      16 = OR (all take two operands except 12 to 14; 17 is string concatenate).

Delimiters are represented by single bytes: 18 = (      These are not strictly needed, but allow the original

19 = )      entered formula to be recreated for editing

20 = ,

@The above 3 need testing@

21 = end of formula

Operands are represented by a byte followed by the value of the operand:

22 = real (8 bytes)

23 = word (2 bytes)

24 = qstr

25 = cell reference

26 = range reference

Functions with no arguments or a fixed list of arguments are represented by single bytes; the operands are treated exactly as for operators. In the following list, the arguments are shown as "n" for numeric, "s" for string, and "r" for range.

27 ERR	55 LOG(n)	83 STRING(n,n)	28 FALSE	56
LOWER(s)	84 CTERM(n,n)			
29 NA	57 MINUTE(n)	85 DATE(n,n)	30 PI	58 MONTH(n)
86 DAVG(r,n,r)				
31 RAND	59 N(r)	87 DCOUNT(r,n,r)	32 NOW	60
PROPER(s)	88 DMAX(r,n,r)			
33 TRUE	61 ROWS(r)	89 DMIN(r,n,r)	34 ABS(n)	62

S(r)	90 DSTD(r,n,r)			
35 ACOS(n)	63 SECOND(n)	91 DSUM(r,n,r)	36 ASIN(n)	64
SIN(n)	92 DVAR(r,n,r)			
37 AT(s)	65 SQRT(n)	93 FIND(s,s,n)	38 ATAN(n)	66
TAN(n)	94 FV(n,n,n)			
39 CELLPOINTER(n)	67 TIMEVALUE(s)	95 HLOOKUP(n,r,n)	40 CHAR(n)	68
TRIM(s)	96 IF(n,n,n)			
41 CODE(s)	69 UPPER(s)	97 INDEX(r,n,n)	42 COLS(r)	70
VALUE(s)	98 MID(s,n,n)			
43 COS(n)	71 YEAR(n)	99 PMT(n,n,n)	44 DATEVALUE(s)	72 ATAN2(n,n)
100 PV(n,n,n)				
45 DAY(n)	73 CELL(n,r)	101 RATE(n,n,n)	46 EXP(n)	74 EXACT(s,s)
102 SIN(n)				
47 HOUR(n)	75 IRR(n,n)	103 TERM(n,n,n)	48 INT(n)	76 LEFT(s,n)
104 TIME(n,n,n)				
49 ISERR(r)	77 MOD(n,n)	105 VLOOKUP(r,n,n)	50 ISNA(r)	78
NPV(n,n)	106 DDB(n,n,n,n)			
51 ISNUM(r)	79 -----	107 REPLACE(s,n,n,s)	52 ISSTR(r)	80
REPEAT(s,n)	108 SYD(n,n,n,n)			
53 LEN(s)	81 RIGHT(s,n)			
54 LN(n)	82 ROUND(n,n)			

Functions with a variable list of arguments are more complex. The call is built up as follows:

- a START byte
  - the arguments
  - an END byte
  - the number of arguments (a byte)
- The arguments are just placed in order. An argument is

either:

- the value of the argument, followed by an ARG byte
- a RANGE byte followed by a range reference

Note that all arguments other than ranges, including cell references, work by evaluating the argument in the normal way; ARG in effect says to pop the next argument off the stack.

The values of the four special bytes differ for each function:

	START	END	ARG	RANGE
AVG()	120	112	136	128
CHOOSE()	121	113	137	129



Type 5 (real formula):

Offset 6 (word): index of formula

Offset 8 (real): current value

Type 5 (text formula):

Offset 6 (word): index of formula

Offset 8 (qstr): current value  
The index of a formula is its position in the set of type 1 records. So 0

means the first type 1 record, 1 the second type 1 record, and so on.

The font byte is simply a number from 0 to 3, giving the font of the cell.

Record type 3 describes column widths, and appears for each column not of the default width. The record takes the form:

Offset 0 (byte): column number

Offset 1 (byte): width

Record type 4 describes the default column width:

Offset 0 (word): default width

Record type 5 holds general status information:

Offset 0 (byte): flags

Bit 0: set if automatic recalculate is on Bit 1: set if protection override is on

Bit 2: set if a cell has been deleted since the last recalculation Bit 3: set if table recalculation is on

Bits 4 to 7: unused, always zero

Offset 1 (byte): unused, always zero

Offset 2 (byte): default numeric display format Offset 3 (byte): default alignment for new cells

The default numeric display format uses the same encoding as cells do, with bit 7 always being zero. The default alignment uses the same encoding as the flags of a cell, with bits 0 to 2, 6, and 7 always being zero.

Record type 6 holds information about the current state of the display: Offset 0 to 7: range reference of titles Offset 8 to 11: cell reference of top left displayed cell excluding titles

Offset 12 to 19: range reference of selected range Offset 20 to 23: cell reference of cursor

Offset 24 (byte): non-zero if grid lines are to be displayed, zero if not Offset 25 (byte): zero if zero values are to be displayed, non-zero if not

Record type 7 describes a named cell or range:

Offset 0 (cstr): name

Offset 16 to 23: range reference

Offset 24 (word): type: 25 = cell, 26 = range

Record type 8 describes a range to be offered for selective printing: Offset 0 to 7: range reference

There may be any number of such records.

Record type 9 describes the criterion and database ranges for the database commands. If this record occurs more than once, the last one will be used.

The record takes the form:

Offset 0 to 7: criterion range reference

Offset 8 to 15: database range reference

Record type 10 describes information for the table commands. If this record occurs more than once, the last one will be used. The record takes the form: Offset 0 to 7: table range reference

Offset 8 to 11: input cell 1 reference Offset 12 to 15: input cell 2 reference, or \$FFFF if only one input cell

Record type 11 describes the print setup. It should not appear in Series 3 spreadsheets - it was used by the MC. The record takes the form:

Offset 0 (byte): flags Bit 0: set if values are shown, clear if formulae are shown

Bit 1: set if hidden cells are shown Bit 2: set if column separators are shown

Bit 3: set if headers are shown

Bits 4 to 7: unused, always zero

Offset 1 (byte): unused, always zero

Record type 12 describes the printer font. It should not appear in Series 3 spreadsheets - it was used by the MC. The record takes the form:

Offset 0 (byte): flags

Bit 0: set for bold

Bits 1 to 2: unused, always zero

Bit 3: set for double height

Bits 4 to 7: unused, always zero

Offset 1 (byte): unused, always zero

Offset 2 to 17: font name

Record type 13 describes a graph (see Psion-/) :

Offset 0 (cstr): name (see Psion-E) Offset 16 to 23: range reference for data range A (see Psion-R)

Offset 24 to 31: range reference for data range B Offset 32 to 39: range reference for

data range C

Offset 40 to 47: range reference for data range D    Offset 48 to 55: range reference for data range E

Offset 56 to 63: range reference for data range F    Offset 64 to 71: range reference for data range X

Offset 72 to 79: range reference for labels for range A (see Psion-B)    Offset 80 to 87: range reference for labels for range B    Offset 88 to 95: range reference for labels for range C

Offset 96 to 103: range reference for labels for range D    Offset 104 to 111: range reference for labels for range E

Offset 112 to 119: range reference for labels for range F    Offset 120 (byte): format of range A

Offset 121 (byte): format of range B

Offset 122 (byte): format of range C

Offset 123 (byte): format of range D

Offset 124 (byte): format of range E

Offset 125 (byte): format of range F    All the formats (see Psion-L) are encoded in the same

way:

Bit 0:            set if lines are shown

Bit 1:            set if symbols are shown

Bits 2 to 7: unused, always zero

Offset 126 (byte): alignment of label range A

Offset 127 (byte): alignment of label range B

Offset 128 (byte): alignment of label range C

Offset 129 (byte): alignment of label range D

Offset 130 (byte): alignment of label range E    Offset 131 (byte): alignment of label range

F

Label range alignments (see Psion-B) are:    0 = centre, 1 = right, 2 = below, 3 = left, 4 = above

Offset 132 (byte): x-axis scaling (see Psion-A)    Bit 0:            set for manual upper range, clear for automatic upper range    Bit 1:            set for manual upper range, clear for automatic upper range

Bits 2 to 7: unused, always zero    Offset 133 (byte): x-axis format (see Psion-A), as for cell format (record

type 2) but with bit 7 always zero

Offset 134 (real): x-axis lower limit (see Psion-A)    Offset 142 (real): x-axis upper limit (see Psion-A)

Offset 150 (byte): y-axis scaling, as for x-axis    Offset 151 (byte): y-axis format, as for x-axis

Offset 152 (real): y-axis lower limit, as for x-axis    Offset 160 (real): y-axis upper limit, as for x-axis

Offset 168 (byte): graph type (see Psion-F)    0 = scatter, 1 = bar, 2 = pie, 4 = line, 5 = stack-bar

Offset 169 (byte): grid flags (see Psion-F)    Bit 0:            set if horizontal grid lines visible

Bit 1:            set if vertical    grid lines visible

Bits 2 to 7: unused, always zero

Offset 170 (byte): colour, encoding unknown    Offset 171 (byte): ranges enabled (see Psion-R)

Bits 0 to 5: set if ranges A to F respectively are enabled    Bit 6:            set if range X is enabled

Bit 7:            unused, always zero

Offset 172 (byte): range labels enabled (see Psion-B)    Bits 0 to 5: set if ranges A to F respectively are enabled

Bits 6 to 7: unused, always zero

Offset 173 (byte): other flags    Bits 0 and 1: font size: 0 = auto, 1 = small, 2 = big (see Psion-F)

Bits 2 and 3: pie labels: 0 = none, 1 = percent, 2 = values (see Psion-F)    Bit 4:            set for 3D graphs (see Psion-F)

Bit 5:            set if Y-axis title enabled (see Psion-I)    Bit 6:            set if X-axis title enabled (see Psion-I)

Bit 7:            set if titles enabled (see Psion-T)

Offset 174 (word): unused

At offset 176 onwards are 10 cstrs. These are, in order:

title first line

title second line

x-axis title

y-axis title

range A legend

range B legend

range C legend

range D legend

range E legend

range F legend

The titles are limited to 40 characters, and the range legends to 20.

Record type 14 indicates which graph is current:

Offset 0 (word): index of current graph(the index is the count of type 13 records, so 0 means the first type 13 record in the file, and so on).

Record type 15 describes the fonts used. It consists of 4 font records, which correspond to fonts 1 to 4. Each font record has the format:

Offset 0 to 5: unknown

Record type 16 holds information about printer set-up, and is identical to record type 2 in Word files (see WORD.FMT).

Record type 17 describes the printer driver. It is identical to record type 2 in Word files:

Offset 0 (byte): printer driver model number

Offset 1 (cstr): printer driver libraryA printer driver library can support several similar printers; the model number specifies which is selected.

Record types 18 and 19 hold the header and footer text respectively as a cstr.

Record type 20 holds additional information about the screen. It will be ignored if it does not directly precede record type 6. It has the format:

Offset 0 (byte): flags

Bit 0: set if grid labels are shown

Bit 1: set if small font is in use

Bits 2 to 7: unused, always zero

Offset 1 (byte): ignored (always zero)

Record type 22 indicates that the file is encrypted. It must be the first record in the file. It has the format:

Offset 0 to 8: encryption key check value

Offset 9 to 15: copy of offset 9 to 15 Offset 16 to 17: the result of encrypting two zero bytes

Encryption

-----

Spreadsheet files can be encrypted with a password on some systems. If so, this fact is indicated by a type 22 record.

The password is used to generate two 9 byte sequences, called the key value and the key check value; there is no obvious relationship between the two sequences. The key check value is written into the type 22 record, while the key value is used for the actual encryption. The key value is generated

with the system call GenMaskInit; there is no documentation of the algorithm used to generate the check value, and it is not the same as used by Word.

[Note: different passwords may generate the same key value but different keycheck values, or vice versa.]

Encryption is carried out using the system call GenMaskEncrypt: the data section of each record (other than the type 22 record) is encrypted in the order they occur in the file, with offset 16 of the encryption control block being zero before encrypting the first record. The type and length of the records are not encrypted.

---

Revision #1

Created Thu, Jan 24, 2019 10:32 AM by Alex

Updated Thu, Jan 24, 2019 10:32 AM by Alex