

Manual

SED48

Version 1.2

© C. Lugtmeier

CONTENTS

COPYRIGHT AND DISCLAIMER OF WARRANTY.....	3
DESCRIPTION.....	3
SED48 LIBRARY MENU	4
COMMANDS.....	4
<i>SEDM</i>	4
<i>ALLP</i>	4
<i>SELP</i>	4
↑ <i>SST</i>	4
↓ <i>SST</i>	5
<i>MTYP</i>	5
<i>STYP</i>	5
<i>PRPS</i>	5
<i>VALS</i>	5
→ <i>DBS</i>	5
<i>DBS</i> →.....	5
<i>CFGRSED48</i>	5
<i>ABOUTSED48</i>	5
COMMAND REFERENCE.....	6
<i>ABOUTSED48</i>	6
<i>ALLP</i>	6
<i>CFGRSED48</i>	6
→ <i>DBS</i>	7
<i>DBS</i> →.....	7
<i>PRPS</i>	7
<i>MTYP</i>	8
<i>SEDM</i>	8
<i>SELP</i>	8
↑ <i>SST</i>	9
↓ <i>SST</i>	9
<i>STYP</i>	10
<i>VALS</i>	10
REMARKS.....	10
DATABASE STRUCTURE.....	11
GENERAL	11
COMPRESSION	11
SPEED	11
EXAMPLES	12
DATABASE.....	12
<i>Steel sections</i>	12
<i>Standard beams</i>	12
<i>Sheet piles</i>	12
PROGRAMMING.....	13
<i>User functions</i>	13
<i>Browsing</i>	13
INSTALLATION.....	15
CHARACTERISTICS	15
AUTHOR.....	15

COPYRIGHT AND DISCLAIMER OF WARRANTY

All the files of the SED48 library are copyrighted © by C. Lugtmeier. SED48 is distributed in the hope that it will be useful, but THE COPYRIGHT HOLDER PROVIDES THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL THE COPYRIGHT HOLDER BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM.

This version of SED48 is a Giftware release. You may use it as long as you like, but only for non-commercial purposes and only as a private person. Permission to copy the whole, unmodified, SED48 library is granted provided that the copies are not made or distributed for resale (excepting nominal copying fees) and provided that you include on each copy this copyright notice and disclaimer of warranty.

DESCRIPTION

The SED48 library is an application for the Hewlett Packard 48 series calculators. It is designed to be used as a database for Structural and Mechanical Engineering. The database is optimized for defining steel sections, standard beams, etc. However, please note that it is not limited to these uses. The database structure is so that it can be used for a lot of different types of data. SED48 is completely written in SysRPL.

SED48 LIBRARY MENU

Commands

The SED48 library menu consists of the following commands:

1. SEDM
2. ALLP
3. SELP
4. ↑SST
5. ↓SST
6. MTYP
7. STYP
8. PRPS
9. VALS
10. →DBS
11. DBS→
12. CFGRSED48
13. ABOUTSED48

The stack diagrams of these commands are listed in the section COMMAND REFERENCE. Below the use of these commands is explained.

SEDM

This command displays an easy to use interface to the most commonly used commands: ALLP, STYP, ↑SST and ↓SST. See the description of these commands for further details on their use. The menu assembles a menukey for each Main_type available in the database. These menukeys provide four actions, two in the *unshifted* plane and two in the *shifted* plane:

- | | |
|-------------------------|--|
| 1. <i>unshifted</i> | ALLP (with %Sub_type as argument on level 1) |
| 2. <i>unshifted</i> | STYP (with '?' as argument on level 1) |
| 3. <i>left-shifted</i> | ↑SST |
| 4. <i>right-shifted</i> | ↓SST |

Please note that the use of these commands through this menu only does not require the Main_type as an argument! Since there is a menukey for each Main_type this is not necessary. The Main_type argument is provided internally by the menu definition.

ALLP

This command is used to extract *all* Properties of the selected Main_type and Sub_type that are defined in the current database. Depending on the status of userflag 45 and 46 the output is given tagged or untagged and in a list or not.

SELP

This command is used to extract *selected* Properties of the selected Main_type and Sub_type that are defined in the current database. Depending on the status of userflag 45 and 46 the output is given tagged or untagged and in a list or not.

↑SST

This command is used to extract a Sub_Type of a selected Main_type that meets the search criterium. For instance, you could search a Sub_type of a Main_type that has an area of at least (equal to or bigger than) 2500.

↓SST

This command is used to extract a Sub_Type of a selected Main_type that meets the search criterium. For instance, you could search a Sub_type of a Main_type that has a height of at most (equal to or smaller than) 500.

MTYP

This command is used to extract all Main_types that are defined in the current database. Depending on the status of 46 the output is given in a list or not.

STYP

This command is used to extract all Sub_types of the selected Main_type that are defined in the current database. Depending on the status of userflag 45 and 46 the output is given tagged or untagged and in a list or not.

PRPS

This command is used to extract all Properties of the selected Main_type that are defined in the current database. Depending on the status of userflag 45 and 46 the output is given tagged or untagged and in a list or not.

VALS

This command is used to extract all Values of the selected Main_type that are defined in the current database. Depending on the status of userflag 45 the output is given tagged or untagged.

→DBS

This command is used to store a database or to delete the current database (both with confirmation if flag 47 is clear).

DBS→

This command is used to recall the current database.

CFGRSED48

This command provides an easy to use interface to manipulate userflag 45, 46, 47 and 48. Below the settings are explained.

Userflag	45	46	47	48
Clear	tagged results	results in list	confirmation on	no BZ on database
Set	untagged results	results on stack	confirmation off	use BZ on database

ABOUTSED48

This command displays a simple “about” message.

COMMAND REFERENCE

ABOUTSED48

This command displays a small “about” message.

IN level	OUT level
1 -	1 \$about

ALLP

This command yields all property values of the selected Main_type & Sub_type that are registered in the database.

IN level	OUT level
2 “Main_type”	with flag 45 clear & 46 clear:
1 %Sub_type	1 Main_type_Sub_type_TAG: {Property TAG: %Values}
	with flag 45 set & 46 clear:
	1 {%Values}
	with flag 45 clear & 46 set:
	n Property_TAG: %Value_1
	n-1 Property_TAG: %Value_2
	2 Property_TAG: %Value_n-1
	1 Property_TAG: %Value_n
	with flag 45 set & 46 set:
	n %Value_1
	n-1 %Value_2
	2 %Value_n-1
	1 %Value_n

CFGRSED48

This command displays the database configuration menu.

IN level	OUT level
1 -	1 -

→DBS

This command is used to store a new database or to delete the current database. You will be asked to confirm the action if flag 47 is clear.

IN level	OUT level
1 {Database} ¹	1 -
<i>or:</i>	
1 : port : ID ²	
<i>or:</i>	
1 %0 ³	

DBS→

This command is used to recall the database to the stack.

IN level	OUT level
1 -	1 {Database}
	<i>or:</i>
	1 : port : ID

PRPS

This command yields the defined Properties of the selected Main_type.

IN level	OUT level
1 "Main_type"	<i>with flag 45 clear & 46 clear:</i> 1 Main_type_TAG: {Properties}
	<i>with flag 45 set & 46 clear:</i> 1 {Properties}
	<i>with flag 45 set & 46 set:</i> n Property_1 n-1 Property_2 2 Property_n-1 1 Property_n

¹ Database stored in hidden directory.

² Database stored by user in port memory. : port : ID refers to the database (which can be compressed).

³ Database deleted from hidden directory.

MTYP

This commands yields all Main_types of the current database.

IN level	OUT level
1 -	with flag 46 clear: 1 {"Main_types"} with flag 46 set: n "Main_type_1" n-1 "Main_type_2" 2 "Main_type_n-1" 1 "Main_type_n"

SEDM

This command displays the SED48 menu interface to the ALLP, ↑SST and ↓SST commands.

IN level	OUT level
1 -	1 -

SELP

This command yields selected property values of the selected Main_type & Sub_type that are registered in the database.

IN level	OUT level
3 "Main_type"	with flag 45 clear & 46 clear: 1 Main_type_Sub_type_TAG: {Selected_property_TAG: % Values}
2 %Sub_type	
1 {Selected_properties}	
or:	with flag 45 set & 46 clear: 1 {% Values}
3 "Main_type"	
2 %Sub_type	
1 Selected_property	with flag 45 clear & 46 set: n Selected_Property_TAG: % Value_1 n-1 Selected_Property_TAG: % Value_2 2 Selected_Property_TAG: % Value_n-1 1 Selected_Property_TAG: % Value_n with flag 45 set & 46 set: n % Value_1 n-1 % Value_2 2 % Value_n-1 1 % Value_n

↑SST

This commands yields a Sub_type of the selected Main_type that is equal to or bigger than the Property_value that is desired.

IN level	OUT level
3 "Main_type"	with flag 45 clear:
2 Selected_property	3 "Main_type"
1 %Minimum_value	2 %Sub_type
	1 Property_TAG: % Value
	with flag 45 set:
	3 "Main_type"
	2 %Sub_type
	1 % Value

Note: The ↑SST command searches from the top row to the bottom row of the database and returns the first value that matches the search criterium.

↓SST

This commands yields a Sub_type of the selected Main_type that is equal to or smaller than the Property_value that is desired.

IN level	OUT level
3 "Main_type"	with flag 45 clear:
2 Selected_property	3 "Main_type"
1 %Maximum_value	2 %Sub_type
	1 Property_TAG: % Value
	with flag 45 set:
	3 "Main_type"
	2 %Sub_type
	1 % Value

Note: The ↓SST command searches from the bottom row to the top row of the database and returns the first value that matches the search criterium.

STYP

This command yields all Sub_types of the selected Main_type.

IN level	OUT level
1 "Main_type"	with flag 45 clear & 46 clear: 1 Main_type_TAG: {%Sub_types} with flag 45 set & 46 clear: 1 {%Sub_types} with flag 45 set & 46 set: n %Sub_type_1 n-1 %Sub_type_2 2 %Sub_type_n-1 1 %Sub_type_n

VALS

This command yields the defined % Values of the selected Main_type.

IN level	OUT level
1 "Main_type"	with flag 45 clear: 1 Main_type_TAG: [%Values] with flag 45 set: 1 [%Values]

Remarks

48G series owners can also use the parallel list processing capabilities of the G series to extend the usability of the commands above.

DATABASE STRUCTURE

General

The database consists of a list of lists. Each of these lists contains a list as first element and a matrix as second element (see also EXAMPLES).

- Database
 - {
 - Main_type_list_1
 - Main_type_list_2
 - Main_type_list_n-1
 - Main_type_list_n
 - }
- Main_type_list
 - {
 - Property_list
 - Values_matrix
 - }
- Property_list
 - {
 - Main_type_string
 - Property_1
 - Property_2
 - Property_n-1
 - Property_n
 - }
- Values_matrix
 - [[Subtype_1 Value_1 Value_2 Value_n-1 Value_n]
 - [Subtype_2 Value_1 Value_2 Value_n-1 Value_n]
 - [Subtype_n-1 Value_1 Value_2 Value_n-1 Value_n]
 - [Subtype_n Value_1 Value_2 Value_n-1 Value_n]]

Compression

In order to save some memory it is a good idea to use a compressed database. SED48 supports the use of the BZ compressor/decompressor. A database that is stored by the user in port memory can be compressed by the user with BZ. Thus a string like "BZ....." will be stored in the desired port. This all in the assumption of the BZ command⁴ being available on your calculator.

Speed

The speed of the database commands depend mostly⁵ on the way the database is stored!

Order of speed	Storage of database
Fastest	Uncompressed in hidden directory
	Uncompressed in port
	Compressed in hidden directory
Slowest	Compressed in port

⁴ As a rom pointer! A BZ program in a directory or HOME is not allowed.

⁵ The size of the database is of influence too.

EXAMPLES

Database

Steel sections

```
{
  {
    { "HEA" "b" "h" "A" "Iy" }
    [[ 100 96 100 2120 3490000 ]
     [ 200 190 200 5380 36920000 ]]
  }
  {
    { "HEB" "b" "h" "A" "Iy" }
    [[ 100 100 100 2600 4500000 ]
     [ 200 200 200 7810 56960000 ]]
  }
}
```

Also allowed is:

```
{
  {
    { "HEA" b h A Iy }
    [[ 100 96 100 2120 3490000 ]
     [ 200 190 200 5380 36920000 ]]
  }
  {
    { "HEB" h A "Iz" }
    [[ 100 100 2600 1670000 ]
     [ 200 200 7810 20030000 ]]
  }
}
```

This means that you can have different properties (if desired) for any Main type! You can also use string and ID⁶ property definitions in the same list⁷.

Standard beams

Used just like steel sections!

Sheet piles

Used just like steel sections!

⁶ The Main type HAS to be a string though.

⁷ Actually, the properties can be in any format, like arrays, lists etc. Most of the time this is not meaningful though.

Programming

User functions

Below an example is given of how to declare a user-function in order to use the SED library command \rightarrow SELP in equations.

This is done for the property "A" (here supposedly the cross-sectional area of a steel section).

'Area'

```
<<  $\rightarrow$  maintype subtype           @ creation of local variables
  << 45 SF 46 SF                 @ no tags, no lists
    maintype
    subtype
    "A"                         @ property "A"
    IFERR
      SELP
    THEN
      IF
        -55 FC?                 @ last arguments system flag
      THEN
        3 DROPN                 @ drop last arguments
      END
      0                          @ return 0 if  $\rightarrow$ SELP errors
    END
  >>
>>
```

Now you can use the function Area in your equations like this:

'Weight=Area(MT,ST)*0.000785'

You could load this equation into the solver and define MT and ST to your liking⁸. Other properties could be declared exactly the same.

Browsing

If you own a 48G series calculator you could use the choose-box engine to browse through your database. Below an example is given for a User-RPL program BR.PRG which accomplishes such a task. This program is also included in the BR.DIR directory in the package in which SED48 came.

'BR.PRG'

```
<< 0 0 0 0 1 1 RCLF
   $\rightarrow$ 
  mt st l1 l2 c1 c2 f           @ creation of local variables
  << 45 CF 46 CF                 @ tags and lists
    DO
      "Main Type" MTYP
      DUP 'l1' STO
      c1 CHOOSE                  @ main type choose box
```

⁸ Note: when entered directly from the commandline you cannot define the Main type as a string since the user function will not accept this. If this should be possible you can add the sequence: \rightarrow STR DUP SIZE 1 - 2 SWAP SUB after maintype and then enter the Main type as an ID (global variable, with quotation marks).

```

IF
THEN
  l1 OVER POS
  'c1' STO @ store choose position
  'mt' STO
DO
  mt STYP
  OBJ→ SWAP
  DUP 'l2' STO
  c2 CHOOSE @ sub type choose box
  IF
  THEN
    l2 OVER POS
    'c2' STO @ store choose position
    'st' STO
    mt st ALLP
    OBJ→ SWAP
    1 CHOOSE @ properties choose box
    IF
    THEN
      DROP
      mt st ALLP @ properties to stack
      1 1 @ quit browser interface
    ELSE
      0 @ back to sub type choose box
    END
  ELSE
    0 1 @ back to main type choose box
  END
UNTIL
END
ELSE
  1
END
  1 'c2' STO @ reset sub type choose position
UNTIL
END f STOF
>>
>>

```

Note:

Users who have the Metakernel installed should replace the CHOOSE command with the CHOOSE2 command.

INSTALLATION

The SED48 library can be installed in any available port on the calculator. However, it works fastest from a non-covered port (port 0 or 1). To install the SED48 library on your calculator do the following:

- Transfer the library to your calculator and place the library on the stack
- Place the port number you wish to store the library in on the stack
- Press the STO button
- Warmstart the calculator (press ON and C simultaneously) or shut it off and then turn it on again
- Purge the variable which still contains the library

If any of this gives you problems you should read the HP manual!

CHARACTERISTICS

SED48	Description
Version	1.2
Library number	1500 (decimal)
Size (bytes)	3855
Checksum	#15905d
Systemflags used	none
Userflags used	45,46,47,48
Language used	Sys-RPL

AUTHOR

C. Lugtmeier
The Netherlands
c.lugtmeier@hccnet.nl
<http://home.hccnet.nl/c.lugtmeier>