application note

Simplifying Routine Analyses using Spectral SBasic Programming Language—Chlorophyll Analysis

Paul Liberatore

Introduction

Spectral, the software used for the Cintra 10/20/40 series of spectrometers, has a powerful built in programming language called SBasic. This language allows routine measurements, calculations and spectra manipulation to be performed and tailored to the analyst's requirements.

Any ASCII text editor can be used to create an SBasic program. SBasic provides its own editor which can be accessed by using the EDIT command in the BASIC menu in the Spectral software. The editor is a simple ASCII editor which can handle files up to 32 kb in size and supports all clipboard functions. Although more than adequate for the scope of data manipulation required in spectroscopy, if you want to use your own editor change the SBasic Editor section of the software's INI file (e.g. Editor = NOTEPAD.EXE in order to use NOTEPAD instead of the internal editor). A useful facility of the SBasic editor is the automatic renumber function.

To run an SBasic program use the RUN command in the BASIC menu and select the appropriate SBasic program.

Analysis of Chlorophyll

As an example of a simple SBasic Macro, the analysis of chlorophyll will be used.

This macro, which is shown in Table 1, will collect a wavelength scan from a Cintra spectrometer and then print the results of some Chlorophyll calculations to the report page. For information regarding the preparation of the sample prior to analysis, refer to reference 2.

Explanation of the Macro example

This section describes some features of the operation of the *chlorop.bas* program that are specific to SBasic. Note that each line requires a line number. Under the ***Basic*** menu there is a renumber option that simplifies renumbering of lines after editing.

100 Clear

This is a standard Basic command to free all variables and memory. It is good practice to make this the first line of every SBasic program.

The next line is:

110 fileCloseAll

This command closes all open windows. Note that it is derived from the 'File menu command 'CloseAll'.

Most menu commands can be executed in SBasic by using the first four letters of the main menu name with the full name of the menu command appended to it.

Some examples are given below.

Menu	Command	SBasic Command
File	CloseAll	FileCloseAll
File	SaveAs	FileSaveAs
Window	MinimizeAll	WindMinimizeAll
Quant	Ratio	QuantRatio

Some commands also require parameters.

You can get help on a menu command and its SBasic command syntax by highlighting the command in the menu and pressing F1.

120 rem…

130 rem…

The rem command allows explanatory remarks to be inserted in a program. These remarks are ignored when the program runs.

140 gbcuv “Parameter” “OPMOD Wavelength”

150 gbcuv “Parameter” “LXF 630.0"

160 gbcuv “Parameter” “HXF 670.0"

170 gbcuv “Parameter” “SPEED 1000.0"

These lines are used to set the instrument operating mode, the start and end wavelengths and scan speed. Most parameters that affect the operation of the instrument can be set using this format.

The user is now prompted:

190 messages ‘Please ensure that the sample compartment is empty’ -1

This line displays the following dialogue box with the above message and an OK button.

The message is displayed until the user clicks OK or until the number of milliseconds specified by the last parameter has elapsed. In this case a parameter of -1 has been chosen which results in the dialogue remaining open until the OK has been clicked.

Note: The text between the ‘ ‘ can be translated into any language or even lengthened and more detail added to suit the level of expertise of analysts who will be performing the analysis.

The baseline is now collected:

230 gbcuv ‘baseline’ 2

This performs an instrument baseline (using the specified instrument parameters) and stores the resulting data into object two. Most instrument commands found in the instrument menu can be executed by using the same syntax.

The next two lines are standard Basic and are used to create a file name for the scan to be collected into and to create a prompt string to prompt for the sample.

The sample scan is now performed:

360 gbcuv ‘scan’ 1 scanName$ ‘Chlorophyll’

This performs a scan into object 1, saving the result to a file named ***scanName$,*** and labeling the scan as ***Chlorophyll***.

A check is made to see if the object exists:

400 if avail(1) = 0 then goto 780

The **avail()** function is used to check if an object exists and takes an object as a parameter and returns either 0 (no, the object doesn’t exist) or 1 (yes the object does exist).

If it does, then the results must be smoothed:

440 windActivate 1

450 calcSmooth 1 3 5

These are both menu commands. First **wind**ow **activate** object 1, then **calc smooth** object 1 into object 3 using a 5 point smooth.

Now the smoothed scan is saved to disk:

500 fileSaveAs 3 scanName$

This is another menu command. **File SaveAs** object 3 into the file **scanName$.**

Now the results are calculated:

540 A! =11.6*absval(3, 660)
-0.777*absval(3, 642.5)

This is standard Basic code except for the **absval()** function. This function will extract the absorbance value from the specified object at the specified wavelength.

The next few lines perform the rest of the calculations and print the results to the text output screen.

Now the text screen is shown:

730 windActivate -2

740 windRestore

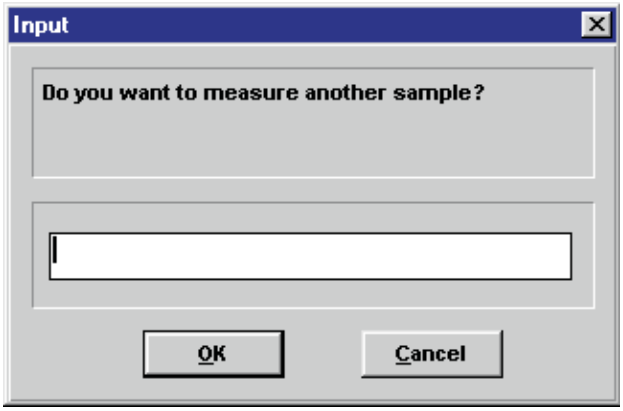
Again the normal menu commands are being used. The special object number -2 is used to reference the text output screen.

Now the user is to be prompted if another scan is required:

790 input 'Do you want to measure another sample', answer$

This is similar to the standard Basic input command but it displays the following dialog with an entry field.

If the user does not answer 'Yes" then the program will end.



Conclusion

The example of the analysis of Chlorophyll (Table 1), demonstrates the power and simplicity of the Sbasic language. The macro ensures that the sample is measured, the calculations are performed and the results are printed all tailored to the analyst's exact requirements.

This macro can be used as a template for writing your own customized applications or can easily be changed to accommodate any local language requirements which you may have.

For convenience, the above Chlorophyll.bas macro can be downloaded from the GBC home page on the Internet. The GBC Home Page can be found on http://www.gbcsci.com.

Of course the Sbasic language can be used to design any customized application.

The macros can be as simple or as complex as required.

References

1. Cintra SBasic and DDE user manual.

2. *Association of Official Analytical Chemists, Official Methods of Analysis.* 15th Edition, Volume 1 pages 62 - 63, 1990.

3. P. G. Aitken. *Microsoft Guide to Visual basic for MS-DOS Programming,* 1992.

4. S. Holzer and the Peter Norton Computing Group. *Peter Norton's Guide to Visual Basic for MS-DOS Programming,* 1992.

5. C. Butkus. *Teach Yourself Visual basic for MS-DOS,* 1992.

```
100  clear
110  fileCloseAll
120  rem  Put the instrument into wavelength scan mode
130  rem  and set the scan range to include 642.5 nm to 660 nm
140  gbcuv “Parameter” “OPMOD Wavelength”
150  gbcuv “Parameter” “LXF 630.0"
160  gbcuv “Parameter” “HXF 670.0"
170  gbcuv “Parameter” “SPEED 1000.0"
180  scan% = 1
190  message “Please ensure the sample compartment is empty” -1
200  rem
210  rem  Perform a baseline into object 2
220  rem
230  gbcuv “baseline” 2
240  rem
250  rem  Set up the scan file name
260  rem
270  scanName$ = “SCAN” + str$(scan%) + “.UVD”
280  rem
290  rem Show prompt
300  rem
310  prompt$ = “Please insert your sample and reference for ” + scanName$  + “ into
       the sample and reference beams”
320  message prompt$ -1
330  rem
340  rem  Perform scan into object 1 saved as ‘SCANxxx.UVD’ and labelled as
‘chlorophyll’
350  rem
360  gbcuv “scan” 1 scanName$  “chlorophyll”
370  rem
380  rem Check that scan was not aborted (is object 1 available?)
390  rem
400  if avail(1) = 0 then goto 780
410  rem
420  rem  Perform a 5 point smooth of the scan and storing the result into object 3
430  rem
440  windActivate 1
450  calcSmooth 1 3 5
460  rem
470  rem  Save object 3 with an ‘S’ appended to the filename
480  rem
490  scanName$ = “SCAN” + str$(scan%) + “S.UVD”
500  fileSaveAs 3 scanName$
510  rem
520  rem  Calculate chlorophyll A for object 3
530  rem
540  A! = 9.93*absval(3, 660.0) - 0.777*absval(3, 642.5)
550  rem
560  rem  Calculate chlorophyll B for object 3
570  rem
580  B! = 17.6*absval(3, 642.5) - 2.81*absval(3, 660.0)
590  rem
600  rem Calculate Total chlorophyll610   total! = 7.12*absval(3, 660.0) +
16.8*absval(3, 642.5)
620  rem
630  rem  Print Results
640  rem
650  print “Result for ”, scanName$
660  print “Chlorophyll A : ”; A!
670  print “Chlorophyll B : ”; B!
680  print “Total Chlorophyll:”; total!
690  print “”
700  rem
710  rem display the text output screen (object number -2)
720  rem
730  windActivate -2
740  windRestore
750  rem
760  rem Ask if another sample is to be measured and increment scan
       number
770  rem
780  scan% = scan% + 1
790  input “Do you want to measure another sample”, answer$
800  answer$ = left$(answer$, 1)
810  if answer$ = “Y” or answer$ = “y” then goto 240
820  end
```

Table 1: *SBasic macro for determination of Chlorophyll.*