Comput. Biol. Med. Vol. 23, No. 2, pp. 131–141, 1993
Printed in Great Britain

0010-4825/93 $6.00 + .00

EXPANDING THE CAPABILITIES OF LABORATORY INSTRUMENTS USING BUILT-IN MICROPROCESSORS AND SERIAL INTERFACES: ADAPTING AN LKB ULTROSPEC® II UV SPECTROPHOTOMETER FOR SCANNING AND ENZYME KINETIC ANALYSES*

JOHN A. BERGES* and CARL VIRTANEN†

* Department of Oceanography, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada; and † Department of Zoology, University of Guelph, Guelph, ON, N1G 2W1, Canada

(*Received* 28 *October* 1992; *received for publication* 15 *December* 1992)

Abstract—An IBM-compatible microcomputer was used to enable an LKB Ultrospec® II UV-Visible spectrophotometer to perform two features which are not available on the basic unit: wavelength scanning and enzyme kinetic analyses. The spectrophotometer was controlled using a simple BASIC program, an RS232c interface and the instrument's own built-in microprocessor. This technique required no special analogue to digital conversion hardware or software and only a rudimentary knowledge of programming. User-designed software permits a greater degree of flexibility in routines and output formats than is offered in most pre-programmed instruments, while offering substantial savings.

Laboratory automation Spectrophotometer Enzyme assays
Wavelength scanning Data acquisition Microcomputers

INTRODUCTION

A wide array of laboratory equipment manufactured today is equipped with an RS232c serial communication port and some degree of on-board microprocessing of data. In many cases, the manufacturer provides optional software, or optional control modules for the equipment. Equally often, however, the user can control the equipment with simple programs and microcomputers. The availability of microprocessors means that the instruments provide data in digital form, avoiding the need for expensive analogue to digital conversion hardware and software [1]. Not only can this route provide a tailor-made means of data-logging and instrument control, but it may also allow the user to obtain features which the manufacturer could provide for the instrument only at substantially greater cost.

In this paper, we present an example using an LKB Ultrospec® II UV-Visible spectrophotometer (LKB Biochrom Ltd, Cambridge, U.K.). By interfacing this unit with an IBM microcomputer and using its built-in microprocessor through an RS232c port, we were able to adapt the unit for routine wavelength scanning and enzyme kinetic analyses, two functions which formerly could not be easily accomplished with this model of the spectrophotometer alone.

DESCRIPTION OF HARDWARE

The LKB Ultrospec® II is a low-cost (less than $10 K Canadian) standard spectrophotometer that allows absorbance or transmittance readings in the ultraviolet and visible ranges from 200 to 900 nm. The standard unit comes with a six-position cell holder, but accessories allowing temperature control and auto-sampling are also available [2]. The spectrophotometer is equipped with a built-in microprocessor and an RS232c serial

communications port, as well as its own instruction set for instrument control and output [2]. The spectrophotometer was connected to an IBM XT computer equipped with 640 kB of RAM. Although not essential for instrument control or data logging, a Colour Graphics Adapter video card was also used for data display. The data connection was made using a null-modem cable to serial communications port 1 (logical DOS device COM1:). In this configuration, pin #2 is used to send data to the computer and pin #3 is used to send instructions to the spectrophotometer. The DIL switches on the spectrophotometer were set to allow data transfer at 9600 baud as described in the instruction manual [2].

DESCRIPTION OF SOFTWARE

Programming language and general considerations

Control programs were written in Microsoft QuickBASIC® (version 4.0), an easily learned and widely available language, which supports serial communication. Only minor adaptations are necessary to allow the programs to run under other versions of the BASIC language (e.g. Borland TurboBASIC, or IBM BASICA). The programs use a CGA graphics adapter (screen 2). Communication is established with COM1: as an input/output device (in this case we assigned the spectrophotometer to device #1) with data transfer at 9600 baud, 8 data bytes, 2 stop bytes and no parity checking. Commands are sent to the spectrophotometer using PRINT commands and the spectrophotometer's instruction set. Each print command must terminate in a line feed and carriage return (ASCII characters 10 and 13). For example, the command "WAVE 300" when printed to device #1 will cause the spectrophotometer to move to a wavelength of 300 nm. To request and collect data from the spectrophotometer, the command "STATUS" is printed to device #1. This results in the output of a 29 character string containing wavelength, absorbance and sample number to the serial port. These data are collected in a character variable using the INPUT command. For example, the commands:

```
PRINT #1, "STATUS", CHR$(10), CHR$(13)
A$ = INPUT(29, #1)
```

will result in wavelength, absorbance and cell number data being retained in variable A$.

Scanning program

A specific application of great usefulness in the laboratory is determining the spectrum of absorbance versus wavelength. This can be useful, for example, in the identification or separation of compounds based on their absorbance characteristics [3]. Often these data must be obtained by laboriously adjusting wavelengths and taking readings manually, or else a graphical plot is made employing some type of X–Y plotter. The latter method may not be as accurate as required if a user must attempt to estimate absorbance values from the plot itself. Alternatively, some spectrophotometers store data and allow them to be analyzed by various "peak finding" software.

We chose to approach the problem in a simple way. The program asks the user to enter (a) the number of samples to analyze (up to 5, plus a reference standard, e.g. the solvent in which the compound is dissolved), (b) the starting and ending wavelength that constitute the scanning range, (c) the interval at which the scan is to proceed (e.g. readings every 1 nm, 5 nm, etc.), and (d) the name of a file to which data will be output. The user is prompted to insert samples, after which the scan commences. For each sample or standard, the spectrophotometer increments the wavelength and records an absorbance, and the user is kept informed of the progress (% complete). Following completion of the scans, data are plotted to the screen and recorded in a data file in ASCII text format. In this form it can be easily imported to a spreadsheet (e.g. Lotus 1,2,3, Lotus Development Corporation) or graphics program (e.g. SigmaPlot, Jandel Scientific) for further analyses. The source code for the program is given in Appendix 1.

To provide an indication of the output, we analyzed two samples of reduced β-nicotinamide adenine dinucleotide (NADH, N4630, Sigma Chemical, St. Louis, MO,

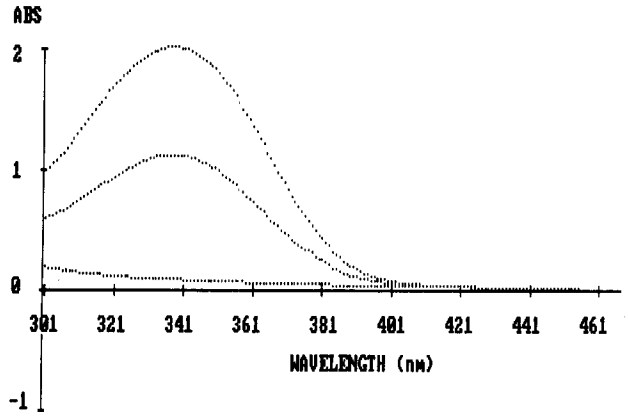

Fig. 1. Screen output of the absorbance versus wavelength plot from the wavelength scanning program. The lowest curve represents data for distilled water alone in a 10 mm path length methylacrylate cuvette. The other curves represent 0.15 and 0.30 mM NADH solutions. Note the characteristic absorbance peak at 340 nm.

U.S.A.) of 0.3 and 0.15 mM concentration. A reference standard of distilled water (the solvent) was also included. Samples were placed in 1.5 ml, 10 mm path length UV grade methylacrylate cuvettes (Terochem Scientific, Markham, Ontario). The scan was performed between 300 and 470 nm at a resolution of 1 nm and took approximately 10 min. This is slow compared to the built-in scanning routines of some spectrophotometers, however, the user may leave the analysis to proceed on its own, and in many cases scanning rate is not an issue. Sample output to the graphics screen is shown in Fig. 1, demonstrating a characteristic peak in NADH absorbance near 340 nm. The file output is given in Fig. 2.

Enzyme kinetics program

A second useful application involves monitoring the rate of an enzyme reaction by recording absorbance over time. For example, many enzymes either oxidize NADH or can be coupled to other reactions which oxidize NADH. Because the product of NADH oxidation (NAD) has virtually no absorbance at 340 nm, recording the change in absorbance over time as the reaction proceeds will give a measure of the reaction rate [4]. Typically, this involves either timing the reaction and taking readings at different intervals (a tedious process), or plotting the data directly using an X–Y recorder (with the inherent problems in estimating values from the plot). In either case, it is difficult to measure more than one reaction at a time.

WAVELENGTH	REFERENCE	SAMPLE 1	SAMPLE 2	SAMPLE 3	SAMPLE 4	SAMPLE 5
300	0.197	0.594	0.987	0.000	0.000	0.000
301	0.192	0.602	1.005	0.000	0.000	0.000
302	0.187	0.612	1.029	0.000	0.000	0.000
303	0.182	0.624	1.056	0.000	0.000	0.000
304	0.176	0.636	1.086	0.000	0.000	0.000
305	0.171	0.650	1.117	0.000	0.000	0.000
306	0.165	0.664	1.150	0.000	0.000	0.000
307	0.161	0.678	1.181	0.000	0.000	0.000
308	0.157	0.694	1.215	0.000	0.000	0.000
309	0.154	0.712	1.252	0.000	0.000	0.000
310	0.152	0.731	1.290	0.000	0.000	0.000
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Fig. 2. Sample data file output from the wavelength scanning program of wavelength, and absorbances for reference solution (distilled water) and 0.15 mM (sample 1) or 0.30 mM (sample 2) solutions of NADH collected in 10 mm path length methylacrylate cuvettes. Note that samples 3, 4, and 5 are unused.

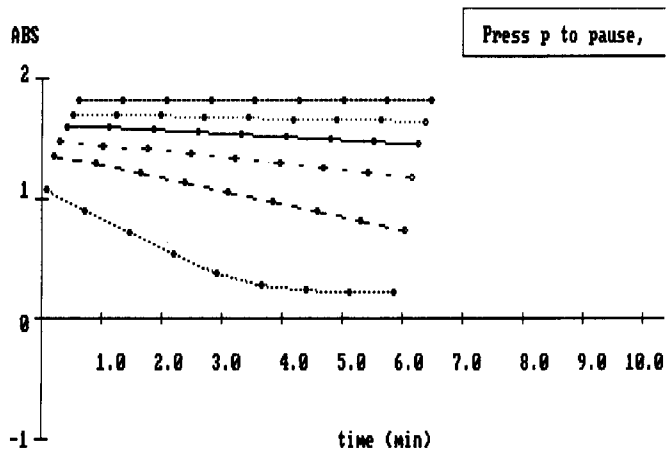

Fig. 3. Screen output of aborbance versus time progress curves from the enzyme kinetics program. From bottom to top, lines represent additions of 50, 20, 10, 5, 1 or 0 mU of lactate dehydrogenase, respectively. Other assay conditions are as described in the text. Note that the screen image was taken only 6 min into the reaction.

Our program prompts the user to enter (a) the desired wavelength to monitor, (b) the number of cells to monitor (up to 6), (c) the length of time to monitor the reaction, and (d) a file to which the data will be recorded. The user may also zero the spectrophotometer using a suitable reference solution. At time intervals between 6 and 45 sec (depending on the number of cells used), each sample is read for absorbance and a paired time and absorbance value recorded. The data are also plotted to the screen (absorbance from -1 to 2.0 relative units versus time in min) to allow monitoring of the progress of the reaction. To prevent data points from falling on top of each other, the results of each sample are offset by 0.1 absorbance units on the plot. During the data collection, the user may abort the data collection (the data file will still be saved), or pause in the course of the reaction to add another substrate. The source code is provided in Appendix 2.

To illustrate the use of the program, we prepared a reaction using the enzyme lactate dehydrogenase (LDH, L2500, Sigma Chemical). The enzyme reduces pyruvate to lactate, oxidizing NADH in the process. Final reaction concentrations were: 50 mM imidazole buffer, pH 7.9 (I0125, Sigma Chemical), 20 mM sodium pyruvate (P2256, Sigma Chemical) and 0.2 mM NADH. The reaction was run in 1.0 ml volumes in 10 mm methylacrylate cuvettes, and was started by the addition of between 1 and 50 mUnits of enzyme (where 1 Unit catalyzes the conversion of one μmol of substrate to product per min). The reaction was monitored for 10 min at 340 nm. Figure 3 shows a screen plot of the reaction at the mid-point. The different amounts of enzyme are reflected in the different slopes of the lines. Figure 4 shows an example of the ASCII file created. Data analysis of this file is easily accomplished through a spreadsheet, utilizing sorting and linear regression routines.

DISCUSSION

These programs provide only a small sample of the possible uses of the combination of built-in microprocessors and microcomputers. Complex analyses can be automated including variations in wavelength, sample number and timing. Prompts can be issued to the user for additions of reagents, allowing less experienced users to perform complex analysis. Once a data file has been created the user may go on to program additional routines for data analyses [5]. Unlike equipment with built-in programming, a high degree of flexibility and specificity in methodology is possible.

The principles developed here can be applied to a wide range of modern laboratory equipment. It is worthwhile for the researcher purchasing such equipment to consider

CELL:	TIME(min):	ABSORBANCE:
1	0.0842	+1.159
2	0.1840	+1.332
3	0.2837	+1.351
4	0.3835	+1.379
5	0.4833	+1.375
6	0.5675	+1.391
1	0.6673	+0.967
2	0.8338	+1.261
3	0.9337	+1.316
4	1.0335	+1.360
5	1.1342	+1.369
6	1.2338	+1.389
1	1.3337	+0.782
2	1.5003	+1.183
3	1.6000	+1.280
4	1.7008	+1.341
5	1.8005	+1.360
6	1.9003	+1.389
1	2.0002	+0.606
.	.	.
.	.	.
.	.	.

Fig. 4. Sample data file output from the enzyme kinetic program of cell (sample) number, time (min) and absorbance. Cells 1–6 represent additions of 50, 20, 10, 5, 1 or 0 mU of lactate dehydrogenase, respectively. Other assay conditions are as described in the text.

the availability of serial ports and microprocessors. Otherwise, such options can be added to many laboratory instruments at modest cost.

Availability of program

The software described in this paper is available free of charge. For copies of the executible files and the source code, send a blank diskette of your choice (one 360 KB or 1.2 MB 5.25″ or 720 KB or 1.44 MB 3.5″ floppy disks) in an appropriate self-addressed mailer to the first author.

SUMMARY

The use of the serial interface and built-in microprocessors common in many modern laboratory instruments can increase the flexibility and capability of these instruments. We provide the example of an LKB spectrophotometer which we have interfaced with an IBM microcomputer. By using simple BASIC programs, we have adapted the instrument for wavelength scanning and enzyme kinetic analyses. The programs provide direct plotting of results to the computer screen and data output to ASCII files for further analyses.

Acknowledgements—The authors acknowledge the support of the Natural Science and Engineering Research Council operating grants to J.C. Roff, J. S. Ballantyne and P. J. Harrison. We thank R. R. Hopcroft and D. M. Jones for valuable programming advice.

REFERENCES

1. D. M. Jones, P. J. Harrison, P. J. Clifford, K. Yin and M. St John, A computer-based system for the acquisition and display of continuous vertical profiles of temperature, salinity, fluorescence and nutrients, *Water Res.* **25**, 1545 (1991).
2. Anonymous, Ultrospec® II Instruction Manual, LKB Biochrom Ltd (1985).
3. S. W. Jeffrey, Properties of two spectrally different components in chlorophyll *c* preparations, *Biochem. biophys. Acta* **177**, 456 (1966).
4. E. F. Rossomando, Measurement of enzyme activity, *Meth. Enzym.* **182**, 38 (1990).
5. V. R. Krishnan, C. A. Morbach and Y. W. Brans, AUTOSPEC®: computerized processing of the data output from a spectrophotometer in a biochemical laboratory, *Comput. Biol. Med.* **21**, 199 (1991).

About the Author—JOHN A. BERGES received his B.Sc. and M.Sc. from the Department of Zoology, University of Guelph in 1987 and 1989, respectively. He is currently completing his Ph.D. in the Oceanography Department at the University of British Columbia. His current research interests include the physiology and biochemistry of marine zooplankton and phytoplankton, as well as technologies for estimating and automating the estimation of biological rate processes.

About the Author—CARL VIRTANEN received his honours B.Sc. in Biological Science from the University of Guelph. He is currently pursuing graduate studies in theoretical ecology in the Department of Zoology, University of Guelph. His interests include mathematical ecology, computer modeling and virtual reality.

APPENDIX 1

```
'*************************************************
'*************************************************
'**  LKB ULTROSPEC II: ENZYME VMAX ASSAY PROGRAM  **
'**  ------------------------------------------  **
'*************************************************
'*************************************************
'
'-------------------------------------------------------------------------------
 'This program was written in June of 1989 by Carl Virtanen and John
 'Berges. They reserve all specific rights regarding this program and its
 'distribution.
'-------------------------------------------------------------------------------

'******************
'*  Definitions   *
'******************

 DECLARE SUB delax (wtime!) :    'Subroutine to allow a delay in execution
 DEFINT C-D, I, N, P, X-Y:       'Integer variables
 DIM hlines%(6), PLC(100, 100): 'Dimensioned variables
 TIME$ = "00:00:00":             'Initialize timer

  hlines%(1) = &HCCCC:           'Define patterns of up to 6 graph lines
  hlines%(2) = &HFF00
  hlines%(3) = &HF000
  hlines%(4) = &HFFFF
  hlines%(5) = &H8888
  hlines%(6) = &H7777

 R$ = CHR$(13): L$ = CHR$(10):  'Define carriage control and line feed for output

 '****************************************
 '*  Prompt User for Initial Parameters  *
 '****************************************

 PRINT "ULTROSPEC II CONTROL PROGRAM"
 LOCATE 5, 1: PRINT "PRESS ANY KEY TO CONTINUE"
 GOSUB HOLD

 CLS
 INPUT "ENTER WAVELENGTH: ", WAVE$: PRINT
 INPUT "NUMBER OF CELLS BEING USED (maximum of 6) :", cmax: PRINT
 INPUT "ENTER MAXIMUM TIME (in minutes) :", maxtim: PRINT
 INPUT "ENTER FILENAME FOR DATA OUTPUT (include disk drive and extension): ", PLACE$

'*************************************************************
'*  Convert user-entered time to number of points to collect  *
'*************************************************************

  IF cmax = 1 THEN
        timint = 6 / 60
  ELSEIF cmax = 2 THEN
        timint = 12 / 60
  ELSEIF cmax = 3 THEN
        timint = 18 / 60
  ELSEIF cmax = 4 THEN
        timint = 24 / 60
  ELSEIF cmax = 5 THEN
        timint = 34 / 60
  ELSEIF cmax = 6 THEN
        timint = 40 / 60
  END IF

  poinmx = INT(maxtim / timint)
  POIN = poinmx * cmax

'************************
'*  Input/Output Setup  *
'************************

  OPEN PLACE$ FOR OUTPUT AS #3
  PRINT #3, "    CELL:          TIME(min):          ABSORBANCE:"
  CLS
```

```
    PRINT "TIMER WILL BEGIN AS SOON AS REFERENCE SEQUENCE IS INITIATED"
    LOCATE 5, 1: PRINT "PRESS ANY KEY TO CONTINUE"
    GOSUB HOLD

    CLS
    OPEN "COM1:9600,N,8,2" FOR RANDOM AS #1:       'Set up serial port 1 for spec
    PRINT "SETTING UP"

'************************************
'*  Spectrophotometer Initialization  *
'************************************

    IF VAL(WAVE$) > 350 THEN :              'If wavelength is UV, strike deuterium lamp
       PRINT #1, "DOFF", R$, L$
    ELSE
       PRINT #1, "DON", R$, L$
       CALL delax(30)
       END IF

    PRINT : PRINT "SET REFERENCE IF DESIRED"
    PRINT : PRINT "PRESS ANY KEY TO CONTINUE"

    PRINT #1, "SA. 1, WAVE "; WAVE$, R$, L$:       'Initialize spec on cell #1 at wavelength
    GOSUB HOLD

    CALL delax(5)
    cell = 1
    TIME$ = "00:00:00"
    CLS

    GOSUB INISCR:                                  'Initialize output screen

    GOSUB GRAB:                                    'Collect a data point

    i = 1: p = 1: GOSUB GRAPH

    '************************************************************
    '*  Loop to Collect and Plot Absorbance versus Time Data  *
    '************************************************************

    FOR i = 2 TO POIN

      IF cell = cmax THEN
         cell = 0: p = p + 1
      END IF

      k$ = INKEY$:                              'Check for keyboard input- q to stop, p to pause
      IF k$ = "q" OR k$ = "Q" THEN
         GOTO FINISH
      ELSEIF k$ = "p" OR k$ = "P" THEN
         LOCATE 1, 40
         PRINT "pausing..... (any key to continue)"
         GOSUB HOLD
         LOCATE 1, 40
         PRINT "                                    "
      END IF

      cell = cell + 1
      GOSUB GRAB
      GOSUB GRAPH

      IF TIMER > 900 THEN
         i = POIN - 1
      END IF

    NEXT i

'************************************************
'*  Routine to Close Files and Terminate Program  *
'************************************************


      PRINT #1, "SAMPLE 1", R$, L$
      CLOSE #1
      CLOSE #3
      LOCATE 22, 1
      PRINT : PRINT "ALL DONE!!!"
    END


    '---------Subroutine to Allow a Pause Until Keyboard Input-------------------
    HOLD:
        DO
        LOOP WHILE INKEY$ = ""
        RETURN

    '---------Subroutine to Collect a Data Point---------------------------------
```

```
GRAB:
    B$ = "SAMPLE " + CHR$(cell + 48)
    PRINT #1, B$, R$, L$:                          'Move to next cell
    CALL delax(5)
    BEFORE = TIMER
    PRINT #1, "::::::::::STATUS", R$, L$:          'Request spec data
    after = TIMER
     a$ = INPUT$(29, #1)
    tm = ((BEFORE + after) / 2) / 60:              'Calculate time of data collection
    sorb = VAL(MID$(a$, 14, 5)):                   'Trim spec data to provide absorbance only
    PRINT #3, USING "      #              ##.####                +##.###"; cell, tm, sorb
    RETURN

'---------Subroutine to Graph a Data Point-----------------------------------
GRAPH:
     IF i > cmax THEN
       x1 = PLC(cell, 1)
       Y1 = PLC(cell, 2)
       PSET (x1, Y1), cell * 2 + 1
    END IF
    Y1 = 137 - INT(sorb * 50) - (cell * 5):        'Scale absorbance data, offsetting each
cell 0.1 units
    x1 = 25 + INT(tm * 550 / (poinmx * timint))    'Scale time data

    IF i > cmax THEN
       LINE -(x1, Y1), cell * 2 + 1, , hlines%(cell):   'Connect data point with a
different type of line for each cell
       CIRCLE (x1, Y1), 2, cell * 2 + 1
    ELSE
       PSET (x1, Y1), cell * 2 + 1
       CIRCLE (x1, Y1), 2, cell * 2 + 1
    END IF

    PLC(cell, 1) = x1
    PLC(cell, 2) = Y1
RETURN

'---------Subroutine to Initialize Screen: Screen 2 = CGA--------------------
INISCR:
    SCREEN 2, 0

    LINE (25, 37)-(25, 187), 2:                    'Draw X and Y axes
    LINE (25, 137)-(600, 137), 2

    interval = INT(500 / 10):                      'Label and Scale Axes

    FOR i = 1 TO 10
       x1 = 25 + INT(i * 50)
       LINE (x1, 139)-(x1, 135), 2
       xlabel! = (i * (maxtim / 10))
       xtikloc = INT((3) + i * interval / 640 * 80)
       LOCATE 20, xtikloc: PRINT USING "##.#"; xlabel!
    NEXT i

    LINE (20, 37)-(30, 37), 2
    LINE (20, 87)-(30, 87), 2
    LINE (20, 137)-(25, 137), 2
    LINE (20, 187)-(30, 187), 2

    LOCATE 24, 35: PRINT "time (min)"
    LOCATE 4, 1: PRINT " 2"
    LOCATE 10, 1: PRINT " 1"
    LOCATE 17, 1: PRINT " 0"
    LOCATE 23, 1: PRINT "-1"
    LOCATE 2, 50: PRINT "Press p to pause,  q to quit"
    LINE (375, 0)-(630, 20), , B
    LOCATE 2, 1: PRINT "ABS"
RETURN

'---------Subroutine to Allow a Specified Delay in Execution-----------------
SUB delax (wtime)

 deltime = TIMER + wtime
   WHILE TIMER < deltime
   WEND

END SUB

'------------------------------------------------------------------------------
```

APPENDIX 2

```
'*****************************************
'*****************************************
'**  LKB ULTROSPEC II SCANNING PROGRAM  **
'**  ---------------------------------  **
'*****************************************
'*****************************************

'-------------------------------------------------------------------------
'This program was written by John Berges for the LKB ULTROSPEC II.
'The program scans reference and samples for a user-specified range
'of wavelengths, plots the results to the screen and creates a data
'file.
'-------------------------------------------------------------------------

'******************
'*  Definitions   *
'******************

DECLARE SUB delax (wtime!) :            'Subroutine to allow a delay in execution
DEFINT C-D, I, N, P, X-Y:               'Integer variable
DIM graf(1 TO 6, 200 TO 900):           'Dimensioned variable
TIME$ = "00:00:00":                     'Initialize timer

r$ = CHR$(13): l$ = CHR$(10):           'Define carriage control and line feed for output

'*************************************
'*  Prompt User for Initial Parameters  *
'*************************************

PRINT "ULTROSPEC II SCANNING PROGRAM"
LOCATE 5, 1: PRINT "PRESS ANY KEY TO CONTINUE"
GOSUB HOLD

CLS
INPUT "ENTER NUMBER OF CELLS (including reference): ", ncells: PRINT
INPUT "ENTER STARTING WAVELENGTH (lower): ", waves$: PRINT
INPUT "ENTER ENDING WAVELENGTH (higher): ", wavee$: PRINT
INPUT "ENTER WAVELENGTH INCREMENT (integer, i.e. 1): ", wincr: PRINT
INPUT "ENTER FILENAME FOR DATA OUTPUT (include disk drive and extension): ", PLACE$

'***********************************************
'*  Sample, Spectrophotometer and File Set-Up  *
'***********************************************

CLS
PRINT "PLACE REFERENCE IN CELL 1, SAMPLES IN CELLS 2 TO "; ncells
LOCATE 5, 1: PRINT "PRESS ANY KEY TO CONTINUE"
GOSUB HOLD

CLS
OPEN PLACE$ FOR OUTPUT AS #3

OPEN "COM1:9600,N,8,2" FOR RANDOM AS #1:            'Set up serial port 1 for spec
PRINT "SETTING UP"

IF VAL(waves$) > 350 THEN :             'If wavelength is UV, strike deuterium lamp
    PRINT #1, "DOFF", r$, l$
ELSE
    PRINT #1, "DON", r$, l$
    CALL delax(30)
END IF

PRINT #1, "SA. 1, WAVE "; waves$, r$, l$:       'Initialize spec om cell #1
CALL delax(7)

'**********************
'*  Scanning Routine  *
'**********************

PRINT "BEGINNING SCAN: Reference"

FOR sampcount = 1 TO ncells
    b$ = "SAMPLE " + CHR$(sampcount + 48)
    PRINT #1, b$, r$, l$
    CALL delax(5)

    FOR waveincr = VAL(waves$) TO VAL(wavee$) STEP wincr
        wave2$ = STR$(waveincr)
        PRINT #1, "WAVE "; wave2$, "::::STATUS"; r$, l$
        CALL delax(2)
        a$ = INPUT$(29, #1)
        sorb = VAL(MID$(a$, 14, 5))
        graf(sampcount, waveincr) = sorb
        percomp = (waveincr - VAL(waves$)) / (VAL(wavee$) - VAL(waves$)) * 100
        LOCATE 5, 5: PRINT percomp; "% COMPLETE"
    NEXT waveincr
```

```
    CLS
    PRINT "BEGINNING SCAN: Sample"; sampcount
 NEXT sampcount

 GOSUB INISCR:                'Initialize output screen

 GOSUB PLOT:                  'Plot data to screen
 GOSUB HOLD
 GOSUB FILEIT:                'Print data to file
 PRINT "Scan complete. Press any key to exit."
 GOSUB HOLD

'************************************************
'*  Routine to Close Files and Terminate Program  *
'************************************************

PRINT #1, "DOFF", r$, 1$
PRINT #1, "SAMPLE 1", r$, 1$
CLOSE #1
CLOSE #3
LOCATE 22, 1
PRINT : PRINT "ALL DONE!!!"
END


'----------Subroutine to Allow a Pause Until Keyboard Input-----------------
HOLD:
    DO
    LOOP WHILE INKEY$ = ""
    RETURN

'----------Subroutine to Initialize Screen: Screen 2 = CGA-------------------
INISCR:
    SCREEN 2, 0

    LINE (25, 25)-(25, 175), 2:                'Draw X and Y axes
    LINE (25, 125)-(600, 125), 2

    swave = VAL(waves$):                       'Scale wavelength data
    ewave = VAL(wavee$)
    iwave = INT((ewave - swave) / 10)
    FOR i = swave TO ewave STEP iwave
       stemp2! = ((i - swave) / (ewave - swave)) * 575!
       X1 = 25 + INT(stemp2!)
       LINE (X1, 127)-(X1, 123), 2
       LOCATE 18, (2 + INT((X1 - 25) / 575 * 72)): PRINT (i)
    NEXT i

    LINE (23, 25)-(27, 25), 2:                 'Scale Absorbance Data
    LINE (23, 75)-(27, 75), 2
    LINE (23, 125)-(27, 125), 2
    LINE (23, 175)-(27, 175), 2
    LOCATE 2, 1: PRINT "ABS"
    LOCATE 4, 1: PRINT "2"
    LOCATE 10, 1: PRINT "1"
    LOCATE 16, 1: PRINT "0"
    LOCATE 22, 1: PRINT "-1"
    LOCATE 18, 30: PRINT "WAVELENGTH (nm) "


 RETURN

'----------Subroutine to plot absorbance vs. wavelength graph----------------
PLOT:
   FOR plotnum = 1 TO ncells

     FOR npoints = swave TO ewave STEP wincr
       temp1 = (npoints - swave) * 575! / (ewave - swave)
       xpoint = 25 + INT(temp1)
       ypoint = 125 - graf(plotnum, npoints) * 50
       PSET (xpoint, ypoint)
     NEXT npoints

    NEXT plotnum


RETURN

'----------Subroutine to record data to a file-------------------------------

FILEIT:

  PRINT #3, "WAVELENGTH     REFERENCE  SAMPLE 1  SAMPLE 2  SAMPLE 3  SAMPLE 4  SAMPLE 5"
  PRINT #3, "----------     ---------  --------  --------  --------  --------  --------"
  PRINT #3,
```

```
  FOR countout = swave TO ewave STEP wincr
     PRINT #3, USING "   ###           #.###     #.###     #.###     #.###     #.###
#.###"; countout, graf(1, countout), graf(2, countout), graf(3, countout), graf(4,
countout), graf(5, countout), graf(6, countout)
  NEXT countout

RETURN

'------------Subroutine to Allow a Specified Delay in Execution--------------
SUB delax (wtime)

DEFSNG C-D, I, N, P, X-Y
 deltime = TIMER + wtime
   WHILE TIMER < deltime
   WEND

END SUB
'-------------------------------------------------------------------------------
```