CHAPTER 4
SOFTWARE FEATURES
4.1 INTRODUCTION

Of the many Programming languages that are currently in common use, none is easier to learn or to use than BASIC language. BASIC was developed in the early 1960's by professors John G. Kemeny and Thomas E. Kurtz. BASIC was designed as a simple, easy-to-use programming language. Yet this remarkably simple language contains enough power and versatility to be used in many different occupations for a wide variety of applications. Over the years, dozens of different versions of BASIC have been developed. Microsoft BASIC is a variation of BASIC that is widely used with personal computers.

Another reason for BASIC's popularity is its widespread availability. This Language is now available on practically every large computer, and it is supported by virtually all commercial time sharing services. Moreover, in recent years BASIC has become the standard language for most
computer applications. Hence, this language can be used on both large and small computers, in all types of different programming environments.

4.2 BASICS OF BASIC LANGUAGE

BASIC is an "User-Friendly" language whose instructions resemble elementary algebraic formulae augmented by certain English keywords such as REM, LET, READ-DATA, GOTO, IF-THEN-ELSE etc. Most other high level languages though have similar features they are generally more complicated, which makes them more difficult to learn and use.

High-level computer languages are often distinguished as being either compiled or interpreted languages. A compiled language program consists of the source code and the compiled code. The source code consists of the program statements in their original form. An interpreted language consists of only source code. The source code is translated line by line directly into machine language instructions. The Microsoft BASIC language is a standard interpreted language used on the IBM PC AT.

4.3 PROGRAM STRUCTURE OF BASIC LANGUAGE

The commands which we give to the computer in BASIC language are called statements and the set of statements given to perform a specific task is called a Program.
Every line in BASIC program should possess the following structure or syntax.

Syntax : < Line No > < Command/Instruction >

The following are some of the rules for writing programs in BASIC language.

(i) Basic program is made up of a set of lines or statements.

(ii) Every line starting with a new statement should have line number at the beginning.

(iii) The line numbers should always be positive integers i.e. they should not be negative integers or fractional number.

(iv) The statements should always be numbered in ascending order and usually in the multiples of 10, so that there is enough space for inserting new statements in between the old statements if necessary in future.

(v) Every statement should have at least one basic command.

(vi) All the commands have to be written using some standard rules which are collectively called syntax rules. If the proper syntax is not followed given, the computer prints out syntax error.

Some of the important instructions of BASIC used in developing software for the present project are described below.

PRINT : In executing the above statement, the computer is to display on the screen whatever is mentioned after the word PRINT. Each PRINT statement prints on a separate line. The syntax of the statement is given below.

Syntax : LNo PRINT < Constant, Variable or Expression >

Ex : 10 LET X = 5
     20 PRINT X

The value of variable X i.e. 5 will be printed on the screen when the program is executed.
INPUT: The INPUT statement is used to enter numerical or string data into the computer during program execution.

Syntax: LNO < Var1, Var2, ...., Var n>

EX: 10 INPUT N
    20 PRINT N

The same statement also can be written in a message form as

10 INPUT " Enter the value of N "; N
20 INPUT X
30 PRINT N*X-2

DIM: The DIM statement provides a mechanism to reserve storage memory for arrays and to set the upper bounds on the number of elements permitted in each array.

Syntax: LNO DIM Var(subscripts)[,(Var(subscripts))]

Here the variable names in the array and the subscripts specify the amount of storage memory to be reserved and set the upper bounds on the number of elements in the array.

FOR TO STEP ; NEXT: This statement is used to repeat a set of statements located between the FOR statement and the NEXT statement.

Syntax: LNO FOR < Loop variable > = Exp1 to Exp2 STEP Exp3
        LNO -------------------
        LNO NEXT

The number of executions is determined by specifying initial and final values of the running variable. It always ends with a NEXT statement. Step is the increment by which the loop variable will be incremented every time the loop is repeated.

READ-DATA: This statement specifies the variables whose values are to be entered into the importer. The DATA statement is to assign appropriate values to the variables listed in the READ statement.

Syntax: LNO READ < Var1, Var2,...., Var n>
        LNO DATA < Data1, data2 .... datan>
GOSUB ; RETURN : The subroutine can be called at any line number in the main program with the help of GOSUB statement. After the execution of instructions specified in the subroutine the control will always come to main program at the statement immediately following the GOSUB statement. Every subroutine must end with a RETURN statement. This statement transfers the control to the main program.

Syntax : LNo GOSUB N

----
LNo RETURN

INP : The statement INP is used to input the binary word from the specified port.
Syntax : Variable = INP (address)
Ex : D1 = INP (&H120)

OUT : The statement OUT is used to output the value to the specified port in the binary form.
Syntax : OUT Address, Value
Ex : OUT &H121, 25

LINE : BASIC includes a statement, LINE, which can be used in graphics mode to draw either lines or boxes. The syntax for LINE is given below.
Syntax : LINE [(al,bl)]-(a2,b2)[,[color][,B[F]][style]]

The (a1,b1), (a2,b2) represent the beginning and ending coordinates of the line respectively.

COLOR : The color statement is the color number which can be range from 0 to 3. In medium resolution, the color statement chooses the color from the current pallet set. If color is not specified, it will default to 3. In high resolution, a color number 0 indicates black, while a color of 1 indicates white.

The following program illustrates how the command LINE can be used to draw a horizontal line on the display.
Ex : 10 SCREEN 1 : COLOR 4,0 'background 4 and pallet 0
20 CLS
OPEN : The BASIC statement OPEN is used to open a file. OPEN can be used in either of the following configurations with sequential files.

Syntax: OPEN "filename" FOR mode AS [#] file number

The abbreviations in the preceding format can be interpreted as follows.

filename : is the file name

mode : can be any of the following

INPUT - for sequential input. Data can only be read from the file. Data cannot be written to it. A "File not found" error will occur if an attempt is made to open a file for input that does not already exist.

OUTPUT - for sequential output. OUTPUT always causes a new file to be created. If a file already exists with the same filename as that specified in an OPEN statement with the OUTPUT mode indicated, existing data in the file will be erased. The present data will be written to that file from its beginning point.

APPEND - for sequential output mode for diskette files only. APPEND is specified when data is to be added to the end of an existing file. If the file being opened for an APPEND already exists, new data will be written to the end of that file. If that file does not exist, a new output file will be created.

file number : is a file number required number which is used to refer to a file while it is open. It is much easier to refer to a file as #1 than by its file specification.
The following are examples of valid OPEN statements.

```
OPEN "A:TRANS.DAT" FOR INPUT AS #1
OPEN "B:TEST.FILE" FOR APPEND AS #2
```

The following are examples of the use of the alternate format of the OPEN statement.

```
OPEN "O",#1,"B:FILE.DAT"
OPEN "I",#2,"A:VENDOR.DAT"
OPEN "O",#1,"C:CHECKS"
```

CLOSE: The statement CLOSE is used to close the files which are already opened by the OPEN statement.

```
CLOSE [[#]Filenum[,[#]filnum]...]
```

The following are examples of valid CLOSE statements.

100 CLOSE 'Causes all open devices and files to be closed.

200 CLOSE #1,#2 'Causes the files and devices associated with file numbers 1 and 2 to be closed.

WRITE: WRITE # statement is used to write the data to the specified file.

The following are the examples of valid WRITE statements.

```
WRITE# filenum.
```

```
100 WRITE #1,S1,S2
```

4.4 SOFTWARE FEATURES

The software package developed for the present study enables the user to carry out the spectrophotometric analysis of the given sample solutions and for estimating the parameters such as %Transmittance and Absorbance in the wavelength range 340-960 nm. The following functions are to be carried out precisely and accurately with the above software program.
1. To rotate the Grating of the spectrophotometer precisely with a stepping motor by generating appropriate pulse signals in sequence through software as explained in Chapter-3.

2. To enable the data acquisition system to convert the Analog information into Digital information.

3. To set the position of the Grating according to the desired wavelength range required by the user.

4. To scan for the wavelength corresponding to the absorption peak of the given Analyte.

5. To compute and display the data of the analyte and graphically represent its spectrum.

6. To make different functional units of the system work in a proper sequential order.

7. To indicate defects of the hardware if any.

A high level language of Microsoft BASIC is chosen for the software development of the present study. The flowchart of the program is presented in Figure 4.1. Some of the salient features that are considered for developing the software are presented in the following paras.

4.5 FLOW CHART, SOFTWARE AND SALIENT FEATURES

Figure 4.1 shows the flow chart for the PC based Spectrophotometer for the analysis of Transmittance, Absorbance and Graphical representation of the Absorbance versus Wavelength characteristics of the analyte. A detailed listing of the software package for the spectrophotometric analysis is given below.
4.6 SOFTWARE OF PC BASED SPECTROPHOTOMETER

10 KEY OFF : CLS
20 REM ** COMPUTER BASED SPECTROPHOTOMETER * spectro.bas **
30 A = &H120 'Port - A
40 B = &H121 'Port - B
50 C = &H122 'Port - C
60 CR = &H123 'Control Register
70 CW= &H92 'Control Word - Port A,B are In & C as Out
80 OUT CR,CW 'initialize of 8255
90 OUT C,0 'Port - C reset
100 FOR II=1 TO 4 'Stepper motor data
110 READ F(II)
120 NEXT II
130 DATA &h03,&h09,&h0c,&h06
140 DIM TEMP(500), SAMPLE(500), REF(500), UAVEC500)
145 DIM AA$(14), BB$(14)
150 CLS : SCREEN 0,1
160 COLOR 15: LOCATE 12,25 :PRINT"SWITCH ON THE SPECTROPHOTOMETER"
170 COLOR 31,0,0 : LOCATE 14,25 :PRINT "HAVE YOU DONE IT (Y/N)?";
180 Y$ = INKEY$
190 IF Y$ = "y" OR Y$ * "Y" THEN 200 ELSE 160
200 COLOR 31: CLS
210 GOSUB 680 'Initialization
220 COLOR 7 : CLS
230 GOTO 1310 'Menu program
240 REM * DISPLAY PROGRAM *
250 X » 0 : Y = 100
260 GOSUB 390 'A/D conversion
270 D7 = (D6/4)*100 : REF(X) = D6 : WL = R1+(X*1.5)
280 LOCATE 2,70 :PRINT"DO :");PRINT USING "#####.##";D6*1000
285 LOCATE 3,70 :PRINT"WL :");PRINT USING "#####.##";WL
290 Y1 = 150 - ABS(INT(D7*1.2))
300 LINE (X+100,Y)-(X+100,Y1)
310 X = X+1 : Y = Y1
320 S$=INKEY$
330 IF S$ = "s" THEN 380 ELSE 340
340 IF WL = 940 THEN 380 ELSE 350
350 GOSUB 870 ' Stepper motor run
360 REM GOSUB 1450 ' delay
370 GOTO 260 ' continue
380 RETURN
390 REM * DATA INPUT FROM PORT *
400 S = INF(B) 'End of Conversion'
410 S1 = &H40 AND S
420 IF S1 = &H40 THEN 430 ELSE 400
430 D1 = INF(A)
440 D2 = INF(B)
450 O = &H10 AND D2

66
460 POL = &H20 AND D2
470 IF 0 = 16 THEN 480 ELSE 490
480 LOCATE 1,70: PRINT "OVER RANGE" : LOCATE 1,70:PRINT "
490 D3 = (15 AND D2) * 256
500 D4 = D3 + D1
510 D5 = (2 * 2 * D4)/4096
520 IF POL = &H20 THEN 530 ELSE 540
530 D6=-D5 : TC = -1 : GOTO 540
540 D6=+D5 : TC = +1
550 RETURN
560 END
570 REM * TO RECORD THE DATA VALUES *
580 TEMP(I) =D7:T(I) = T :I = I + 1
590 RETURN
600 REM * TO SAVE THE RESULTS TO FILE *
610 OUT C,0
620 OPEN "temp.dat" FOR OUTPUT AS #1
630 FOR J = 0 TO X
640 WRITE #1,340+(J*1.5),REF(J),SAMPLE(J)
650 NEXT J
660 CLOSE #1
670 RETURN
680 REM * STEPPER MOTOR INITIALIZATION *
690 SCREEN 0,1:COLOR 31,0,0:LOCATE 12,33:PRINT"INITIALIZATION":COLOR 7
700 FOR II= 4 TO 1 STEP -1
710 OUT C,F(II)
720 INI = INF(B)
730 INIT = &H80 AND INI
740 IF INIT = &H80 THEN 780
750 FOR JJ = 1 TO 25 : NEXT JJ
760 NEXT II
770 GOTO 700
780 OUT C,0
790 REM * STEPPER MOTOR DRIVE *
800 FOR KK = 1 TO 40
810 FOR II = 1 TO 4
820 OUT C,F(II)
830 FOR JJ = 1 TO 25 : NEXT JJ
840 NEXT II
850 NEXT KK
860 RETURN
870 REM * STEPPER MOTOR RUN *
880 FOR II = 1 TO 4
890 CONTR = SEL OR F(II)
900 OUT C,CONTR
910 FOR JJ = 1 TO 25 : NEXT JJ
920 NEXT II
930 RETURN
940 REM * DELEY AFTER THE MOTOR TURN *
950 FOR DD = 1 TO 5 : NEXT DD
960 RETURN
970 REM * PROGRAM FOR REFERENCE INSERTION *
980 CLS
990 LOCATE 2,10: PRINT "INSERT THE REFERENCE CUVETTE AND 
CLOSE THE SHUTTER"
1000 LOCATE 3,10: PRINT "HAVE YOU DONE IT (Y/N)"
1010 Y$ = INKEY$
1020 IF Y$ = "y" THEN 1030 ELSE 990
1030 A$ = "ref"
1040 RETURN
1050 REM * PROGRAM FOR REFERENCE INSERTION *
1060 CLS
1070 LOCATE 2,10: PRINT "INSERT THE SAMPLE CUVETTE AND 
CLOSE THE SHUTTER"
1080 LOCATE 3,10: PRINT "HAVE YOU DONE IT (Y/N)"
1090 Y$ = INKEY$
1100 IF Y$ = "y" THEN 1110 ELSE 1070
1110 A$ = "sample"
1120 RETURN
1130 REM * STEPPER MOTOR REVERSE *
1140 FOR KK = 1 TO R3
1150 FOR II = 4 TO 1 STEP -1
1160 OUT C,F(II)
1170 FOR JJ = 1 TO 25: NEXT JJ
1180 NEXT II
1190 NEXT KK
1200 RETURN
1210 REM * CONVERSION INTO FILE *
1220 CLS
1230 INPUT "Enter the name of file 1";A$
1240 OPEN "temp.dat" FOR INPUT AS #1
1250 OPEN A$ FOR OUTPUT AS #2
1260 INPUT #1,S1,S2,S3
1270 WRITE #2,S1,S2,S3
1280 IF EOF(1) THEN 1290 ELSE 1260
1290 CLOSE #1,#2
1300 RETURN
1310 REM * MENU PROGRAM *
1320 KEY OFF: CLS
1330 SCREEN 2
1340 FOR I=70 TO 580 STEP 1
1350 LINE(I,145)-(I,150)
1360 NEXT I
1370 LOCATE 4,20: PRINT "COMPUTER BASED SPECTROPHOTOMETER (MODEL CL-27A)"
1380 LOCATE 7,28: PRINT " 1. INITIALIZATION 
1390 LOCATE 9,28: PRINT " 2. TRANSMITTANCE 
1400 LOCATE 11,28: PRINT " 3. ABSORBANCE 
1410 LOCATE 13,28: PRINT " 4. PLOT GRAPH 
1420 LOCATE 15,28: PRINT " 5. PRINT GRAPH 
1430 LOCATE 17,28: PRINT " 6. QUIT 
1440 FOR J=70 TO 580 STEP 1
1450 LINE(J,145)-(J,150)
1460 NEXT J
LOCATE 22,26: INPUT "ENTER YOUR CHOICE ="; M
ON M GOSUB 680, 1500, 1920, 3500, 1920, 1920
GOTO 1310

REM * PERCENTAGE OF TRANSMITTANCE PROGRAM *
CLS
R1 = 340 : R2 = 940
R3 = ABS((R2-R1)/1.5)
R = ABS((R1-340)/1.5)  ' R is no. of steps for motor
CLS
SCREEN 0,1: COLOR 31,0,0 : LOCATE 1,77: PRINT "WAIT"
GOSUB 1840 : CLS  'Motor forward
SCREEN 0,1
COLOR 15:LOCATE 12,30:PRINT "Insert Reference Cuvette & Close Shutter"
COLOR 31,0,0 : LOCATE 14,30: PRINT "Have you done it (Y/N)";
Y$ = INKEY$
IF Y$ = "y" OR Y$ = "Y" THEN 1630 ELSE 1610
GOSUB 1960  'Scale
GOSUB 2070
GOSUB 2140
GOSUB 240  'ADC
SCREEN 0,1: COLOR 31,0,0 : LOCATE 1,77: PRINT "WAIT"
GOSUB 1130 'motor reverse
CLS
COLOR 15:LOCATE 12,30:PRINT "Insert Sample Cuvette & Close Shutter"
COLOR 31,0,0 : LOCATE 14,30: PRINT "Have you done it (Y/N)";
Y$ = INKEY$
IF Y$ = "y" OR Y$ = "Y" THEN 1750 ELSE 1730
GOSUB 1960  'scale format
GOSUB 2400
GOSUB 2140
GOSUB 2510
OUT C,O
GOSUB 600  'write data to file
GOSUB 1210  'Converting into file
GOSUB 700  'initialisation
RETURN
REM * STEPPER MOTOR FORWARD *
FOR KK = 1 TO R
FOR II = 1 TO 4
OUT C,F(II)
FOR JJ = 1 TO 25 : NEXT JJ
NEXT II
NEXT KK
RETURN
REM * QUIT PROGRAM *
CLS
LOCATE 12,30 : PRINT "THANK YOU - SEE YOU AGAIN"
1960 REM * PROGRAM FOR GRAPH SCALE *
1970 CLS : SCREEN 2 : PSET(100,30)
1980 DRAW "D120 R400"
1990 FOR J = 100 TO 500 STEP 50
2000 LINE (J,150)-(J,145)
2010 NEXT J
2020 FOR J=150 TO 30 STEP -24
2030 LINE (100,J)-(105,J)
2040 NEXT J
2050 LOCATE 23,30 : PRINT "WAVELENGTH IN nm"
2060 RETURN
2070 REM * X SCALE *
2080 RR1 = 340
2090 FOR JJ = 12 TO 60 STEP 6
2100 LOCATE 21,JJ : PRINT RR1
2110 RR1 = RR1+75
2120 NEXT JJ
2130 RETURN
2140 REM * Y SCALE *
2150 LOCATE 4,9 : PRINT "100"
2160 LOCATE 7,10: PRINT "80"
2170 LOCATE 10,10 : PRINT "60"
2180 LOCATE 13,10: PRINT "40"
2190 LOCATE 16,10 : PRINT "20"
2200 LOCATE 19,11: PRINT "0"
2210 RETURN
2220 REM * PROGRAM FOR PLOT GRAPH *
2230 CLS
2240 LOCATE 10,20 : PRINT "SELECT TYPE OF GRAPH"
2250 LOCATE 12,20 : PRINT "1. % TRANSMITTANCE"
2260 LOCATE 14,20 : PRINT "2. ABSORBANCE"
2270 LOCATE 16,30 : INPUT "SELECT OPTION = ";M
2280 ON M GOTO 2380, 3000
2290 END
2300 REM * READING OF LABELS *
2320 FOR I = 1 TO 14
2330 READ D$,E$
2340 AA$(I) = D$ : BB$(I) = E$
2350 NEXT I
2370 RESTORE : RETURN
2380 REM * PROGRAM FOR TRANSMITTANCE LABEL *
2390 GOSUB 1960 : GOSUB 2300
2400 FOR I = 1 TO 14
2410 LOCATE 4+I,3 : PRINT AA$(I)
2420 NEXT I
2430 RETURN
2440 END
2450 REM * PROGRAM FOR ABSORBANCE LABEL *
2460 GOSUB 1960 : GOSUB 2300
2470 FOR I = 1 TO 10
2480 LOCATE 6+I,3 : PRINT BB$(I)
REM * PROGRAM FOR SAMPLE *
X = 0 : Y = 0 : Y2 = 100
GOSUB 390 " A/D conversion"
D7 = (D6/REF(X))*100: SAMPLE(X)=D6 : WL = R1+(X*1.5)
LOCATE 2,70 : PRINT "DO : ": PRINT USING "###.##"; D6*1000
LOCATE 3,70 : PRINT "WL : ": PRINT USING "###.##"; WL
Y3 = 150 - ABS(INT(D7*1.2))
Y1 = 150 - ABS(INT(D6*1.2))
LINE (X+100,Y2)-(X+100,Y3)
X = X + 1 : Y = Y1 : Y2 = Y3
S$ = INKEY$
IF S$ = "a" THEN 2650
IF WL = 940 THEN 2650 ELSE 2630
GOSUB 870 ' Stepper motor run
GOTO 2530 ' continue
RETURN
REM * PLOT GRAPH *
CLS
INPUT "Enter name of file = "; A$
OPEN A$ FOR INPUT AS #1
X = X + 1 : INPUT #1, S1, S2, S3
TEMP(X) = S3/S2
IF EOF(1) THEN 3517 ELSE 3514
CLOSE #1
GOSUB 2300 : GOSUB 1960 : GOSUB 2070 : GOSUB 2140
R1 = 340 : Y2 = 100 : X = 0
D6 = TEMP(X) ' From file
D7 = D6*100 : WL = R1+(X*1.5)
LOCATE 2,70 : PRINT "DO : ": PRINT USING "###.##"; D6*1000
LOCATE 3,70 : PRINT "WL : ": PRINT USING "###.##"; WL
Y3 = 150 - ABS(INT(D7*1.2))
Y1 = 150 - ABS(INT(D6*1.2))
LINE (X+100,Y2)-(X+100,Y3)
X = X + 1 : Y = Y1 : Y2 = Y3
S$ = INKEY$
IF S$ = "a" THEN 3650
IF WL = 940 THEN 3650 ELSE 3630
REM ' Stepper motor run
GOTO 2530 ' continue
REM LOCATE 2,5 : PRINT "Press shift + prtSc to Print ",
D$ = INKEY$
IF D$ = "q" THEN 3680 ELSE 3660
RETURN
The software developed for the spectrophotometric analysis, processes data and calculates the parameters such as Transmittance, Absorbance and Graphically displays the spectrum of the given analyte. This package is developed as a "MENU OPERATED" software similar to a standard software package menu. The main menu is presented in Figure 4.2.

4.7 SOFTWARE PROCEDURE ADOPTED TO THE SPECTRUM ANALYSIS

The MENU of the software package developed for the PC based spectrophotometer contains SIX OPTIONS.

The FIRST OPTION of the menu initializes all the Programmable Peripheral Interfaces (PPI's) for data transfer to and from the ports. It also initializes the position of the Grating of the spectrophotometer to the starting wavelength i.e at 340 nm. This is achieved by generating a sequence of pulses to activate the stepper motor drive system to rotate the stepper motor in both forward and reverse directions.

The SECOND OPTION of the menu performs the analysis of the data for calculation of Transmittance and its graphical representation of the Absorption spectrum. The main program for this parameter analysis contains the subprograms of Stepper Motor forward and reverse subroutines which brings the grating to the required position. It also includes screen Format, A/D conversion and Display programs.
1. INITIALIZATION
2. %TRANSMITTANCE
3. ABSORBANCE
4. PLOT GRAPH
5. PRINT GRAPH
6. QUIT

ENTER YOUR CHOICE =?
In %Transmittance analysis for the given analyte, first the analysis is made for blank solution to the required wavelength range and these data values are stored in a file using arrays and its graphical representation is presented on the screen along with the detector output and its corresponding wavelength data at the bottom of the graph. The graphical representation of the reference (Distill water) is shown in Figure 4.3. After the completion of the analysis of the blank solution, immediately the stepper motor reverse program brings back the grating to the starting point. The Computer then asks the user to insert the Sample solution. Then the sample solution is inserted into the detector and the entire scanning, parameter calculation and display are repeated. The analog output of the detector system is converted into digital data using the following equation.

\[
\text{Digital Data output} = \frac{V_{\text{ref}} \times 2 \times (\text{Port Data})}{4096} \quad (4.1)
\]

The A/D data values corresponding to each wavelength are divided by the reference data values stored already in the memory and this value is multiplied by 100 will give the %Transmittance value at the corresponding wavelength and %T can be calculated using the following equation.

\[
\text{Percentage Transmittance} = \frac{\text{Sample Data}}{\text{Reference Data}} \times 100 \quad (4.2)
\]
FIGURE 4.3: GRAPHICAL REPRESENTATION OF REFERENCE (DISTILLED WATER) SOLUTION.
Figure 4.4: Wavelength-%Transmittance Spectrum of Potassium Permanaganate (Sample Solution)
The **THIRD OPTION** in menu performs the function in obtaining the Absorbance-Wavelength characteristics and displays the graphical representation of the given analyte. The analysis process similar as in the case of second option. The Absorbance is calculated through the software using the following equation.

\[ A = -\log_{10}(1/T) \]  

Where \( T \) is the transmittance of the given analyte.

The **FOURTH OPTION** in menu displays the graphical representation of the analyte whose data is already available in a floppy disc or hard disc. When this option is selected a message appears on the screen as Enter the file name. After entering the file name displays the graph of the given analyte.

The **FIFTH OPTION** prints the spectra of the given analyte whose data is already stored in floppy disc or hard disc.

The **SIXTH OPTION** in menu quit the software package to the system prompt.

**4.8 OPERATION OF THE SOFTWARE PACKAGE FOR DISPLAYING THE SPECTRUM OF THE ANALYTE**

The software package developed for the computer based spectrophotometer can be run by inserting the floppy disc containing the software in to the drive system. Then immediately after entering the SPECSOFT, the system software
immediately after entering the SPECSOFT, the system software can be executed and displays a message as

```
SWITCH ON THE SYSTEM
HAVE YOU DONE IT (Y/N)
```

by providing the blinking option of the second message which will caution the user. If the user response is "Y" then immediately the message INITIALISATION with blinking will appears on the screen. After the initialization process is over, a menu appears as shown in Figure 4.2. If the user requires the analysis for \( \% \text{transmittance} \), the corresponding option can be selected by typing its number and press the ENTER key. A message appears on the screen for the user as

```
ENTER WAVELENGTH RANGE (Lower,higher)
```

Then the user has to give the wavelength range values through the keyboard and press the ENTER key. A WAIT message with blinking appears on the screen if the lower wavelength value is more than 340 nm. After this again a message will be given by the computer as

```
INSERT THE REFERENCE CUVETTE AND CLOSE THE SHUTTER
HAVE YOU DONE IT (Y/N)
```

If the user's response is "Y", then the computer accepts the data coming from the 12-Bit A/D converter of the detector output of each wavelength and displays the same.

After the completion of the process for the reference solution again a message "WAIT appears on the screen as in the previous case during which time the grating is brought
back to the initial wavelength position as desired by the user at the beginning of the analysis. After this a message appears on the monitor.

**INSERT THE SAMPLE CUVETTE AND CLOSE THE SHUTTER**

**HAVE YOU DONE IT (Y/N)**

The same process will be repeated and the data will be stored in memory. At the same time the transmittance-Viz spectrum %Transmittance-Wavelength characteristics or the Absorbance-Wavelength Spectra are displayed depending upon the option selected in the menu by the user. After the completion of the process for both the reference and sample solutions, the computer displays a message to store the data in to floppy disk or hard disk with the appropriate name of the file. Then a message appears on the monitor as

**ENTER THE FILE NAME**

By entering the file name of the given analyte, the corresponding data can be stored into the memory storage which contains the data of wavelength, Reference and Sample solution data which can be used for further analysis. After saving the file, the computer again displays the menu for the analysis of another sample. If the user wants to come out from the software package operation, sixth option can performs this function. The graphical display of the required analyte can be achieved on the screen with the help of fourth option of the menu.