MICROCOMPUTER INTERFACING WITH
CHEMICAL INSTRUMENTATION

by

Joel R. Meyer

****

Submitted in partial fulfillment
of the requirements for
Honors in the Department of Chemistry

UNION COLLEGE

May, 1982

[illegible]
82
M612
1982

Special thanks to T.C. Werner and L. Hull for their help and support throughout this project.

Dedicated to Regula Frey

TABLE OF CONTENTS

MEYER, JOEL Microcomputer Interfacing With Chemical Instrumentation. Department of Chemistry May 1982.

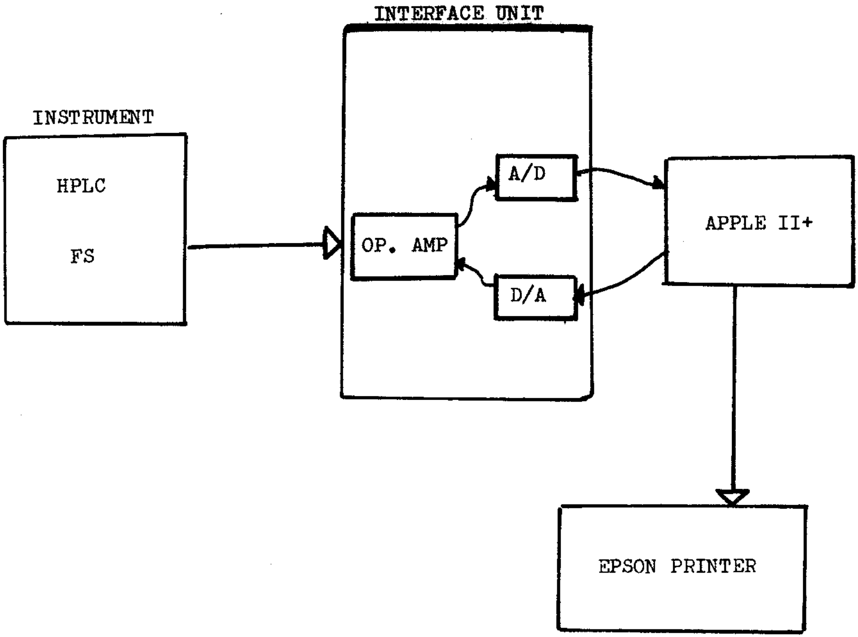
In this paper we discuss the interface of a liquid chromatograph with an Apple II+ microcomputer. The system conists of a Varian 5000 High Performance Liquid Chromatograph (HPLC), a Tracor 970A variable wavelength absorption detector (VWD), an Apple II+ computer, and an interface unit which consists of an operational amplifier and a Mountain Computer A/D + D/A converter. The op. amp. modifies the output signal of the VWD (0 to +1V) to meet the input specifications of the A/D converter (-5V to +5V). This allows the conversion of the continuous analog signal from the VWD to a digital signal, which is required by the Apple. The system produces chromatograms (digital absorption equivalent value vs. time) using the high resolution graphics screen of the Apple, which can be printed with an Epson dot matrix printer. Further, it calculates retention times, performs peak integration, and prints out the results from these operations. A comparison of the Apple's integration with standard cut-and-weigh as well as triangle approximation methods gives an indication of the reliability of the integration technique.

A manual, which requires that the user have only minimal computer experience, has been written to facilitate user access to the system.

This project involves interfacing an Apple II+ microcomputer with two different laboratory instruments: a Varian 5000 High Performance Liquid Chromatograph (HPLC) and a Perkin-Elmer MPF-2A Fluorescence Spectrophotometer (FS). The interface systems are quite similar, in that both involve modifying an analog signal to meet the input specifications of Mountain Computer's A/D + D/A converter, which converts the analog signal to a digital signal. Using Apple BASIC and assembly language, we can store the digital signal in successive memory locations of the Apple as a function of time. The digital absorbance equivalent signal may then be plotted as a function of time for the HPLC and the digital fluorescence equivalent as function of wavelength for the FS, to produce a chromatogram and a fluorescence spectrum, respectively. In addition, the system performs relative and absolute integrations and makes hard copies of the processed data using an Epson dot matrix printer.

The schematic below (Figure 1) represents the pathway of a signal from the real world through the interface unit, which consists of the necessary operational amplifiers and the A/D converter, and into the Apple. This paper will explain the different steps in the modification and processing of the data signal

Figure 1 - Schematic



obtained from the various instruments. Parts of the inteface system will be analysed in the order that the signal passes through them, except for the A/D + D/A converter. An understanding of this unit is necessary in order to follow the design considerations for the op. amps. For this reason, it is appropriate to consider this part of the system after the description of the apparatus in section 1. In addition, since most of the analysis is quite similar for the two instruments, the bulk of the analysis will be done with respect to the HPLC, and the essential considerations for the FS system will be dealt with in a separate section.

Section I: Instrumentation

A) HPLC

The HPLC setup consists of a Varian 5000 HPLC and a Tracor 970A variable wavelength absorption detector (VWD). The VWD has an analog output, in addition to the output used for the recorder, which is specifically designed for interfacing. The range of the this output is from 0 to 1V. The analog signal is not a function of the Absorbance setting on the TRACOR 970A, which is commonly used to amplify the output signal to the recorder. This is convenient, because the signal to the HPLC op. amp. will thus always be confined to the same range. This simplifies the design of op. amps. needed to modify the analog signal to the specifications of the A/D converter. The design of the op. amp. will be discussed after the discussion of the A/D + D/A converter.

B) Fluorescence Spectrophotometer

This instrument consists of a xenon power supply, a power unit, a spectrophotometer, and a recorder. An output analog signal is found on the back of the spectrophotometer, but, unlike with the HPLC, this signal's range is a function of the Sensitivity setting (analogous to the absorbance setting on the HPLC) on the spectrophotometer power supply. Instead this range varies with changes in the sensitivity setting, since these range changes are controlled by switching resistors in the feedback loop of an amplifier in the spectrophotometer power supply. It is possible to construct a circuit that has a constant output range at the different sensitivity settings. The circuit contains a five pole switch which offsets the changing sensitivity adjustment on the power supply. Further, it is advantageous to design this circuit so that the output range is 0 to 1V. This way, the op. amp. designed for the HPLC can be used for the final modification of the output analog signal, making the range acceptable to the input specifications of the A/D converter. The design for the fluorometer's first op. amp., as well as the HPLC's op. amp., will be shown in the section on operational amplifiers.

C) A/D + D/A Converter

As mentioned previously, it is important to understand the specifications for this piece of hardware in order to understand the design strategy of the operational amplifiers. It is for this reason that we discuss this part of the interface prior to a discussion of op. amps.

Mountain computer's A/D + D/A board can be purchased as a unit, which includes the actual board and two blue twenty five pin ribbon cables. The unit fits conveniently into one of the slots in the Apple. Installation is relatively straightforward and instructions are included in the A/D + D/A operating manual(1), which comes with the unit. In addition, Mountain Computer supplies a demonstration diskette which consists of several sample applications and a self test for the board. Running the self test program determines whether or not all parts of the A/D + D/A board are "functional and properly installed." A complete description of Self Test can be found in the A/D + D/A manual(2).

The A/D + D/A board functions within a specific voltage input range. The board converts a continuous analog signal, from -5V to +5V, to a discrete digital signal, ranging from 0 to 255. At input voltages outside this range, the board gives the upper or lower limit of the digital scale depending on whether the analog signal is greater than +5V or less than -5V. Below is the chart (Table 1) included in the A/D + D/A manual, which illustrates A/D as well as D/A conversion.

It is important to note that the digital values produced by the ADC are subject to certain performance factors. That is, the board is functioning properly as long as the absolute deviation for the full scale reading is within +/- 3% and the relative deviation is +/- 1 for the least significant bit. Also worthy of mention is the fact that the ADC only uses addresses in memory between $C090 and $C0FF. These addresses, expressed in hexadecimal code, correspond to a range of 49,296 to 49,407 in decimal code. The significance of this will be explained in the section on memory usage. Finally, the conversion from analog to digital requires nine microseconds. This represents the fastest time that an analog signal can be monitered using the ADC.

The unit also includes two twenty five pin ribbon cables which are used as the input and the output of the A/D and D/A converter. The pin plan included in the manual for these two cables is incorrect;

Table 1 - A/D + D/A Conversion Chart

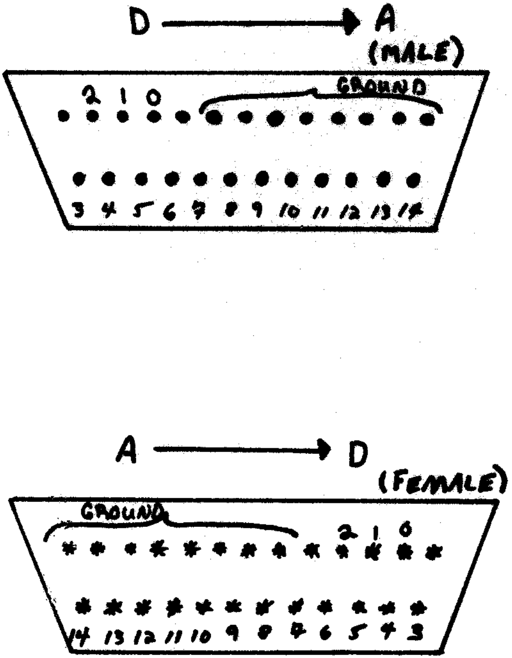
A/D + D/A Conversion Chart

(Output) DAC	Number	(Input) ADC
-5.00 V	0	-5.00 V
	• •	
-2.50 V	64	-2.50 V
	• •	
0.00 V	128	0.00 V
	• •	
+2.50 V	192	+2.50 V
	• •	
+5.00 V	255	+5.00 V

therefore, we show the corrected plan in Figure 2. It is worthy of note that each plug has 15 channels (0-14) for conversions, 8 grounds, and 2 nonsense channels. Furthermore, the plugs are mirror images of each other, which greatly simplified the correction of the outdated pin plan.

The manual for the A/D + D/A board also includes instructions on how to read a particular channel for the ADC and how to program a particular channel for the DAC. Since these instructions would be more useful when presented with the software for data collection, the analysis is not repeated here.

Figure 2 - Corrected Pin Plan



D) Operational Amplifiers

1) HPLC

In general, there are three important rules to remember when working with op. amps.:

1)They have a high input impedance as well as a low output impedance.

2)They always act to keep the potential difference between their two inputs, inverting and non-inverting, equal to zero.

3)Their output voltage depends only on the external connections to the op. amp.

With these rules in mind, we proceeded to make the output signal from the VWD compatable with A/D converter's input specifications. Recall that the optimum voltage range for the ADC is -5V to +5V, and that the output of the VWD varies from 0V to 1V. The modification, therefore, involves an amplification as well as an offsetting of the output signal from the VWD. More precisely, the output signal from the VWD must be amplified by a factor of 10 and offset with a constant -5V. This would convert the output voltage range of the VWD to the optimum input voltage range for the ADC. An op. amp. circuit that performs this conversion is shown in Figure 3. The circuit consists of two op. amps. and various resistors. The first op. amp. has two input voltages that intersect at the inverting input of amp. 1. These are the output voltage from the VWD, E(i), and an offset voltage of -5V from the DAC. The output voltage of the first op. amp., E(1), depends on the external connections to the amp. in the following manner:

$$E(1)= -(E(i)(R3/R1) +(-5V)R2/R1) \qquad (1)$$

This can be simplified by plugging in the values of the various resistors as follows:

$$E(1) = -(10E(i) - 5V) \qquad (2)$$

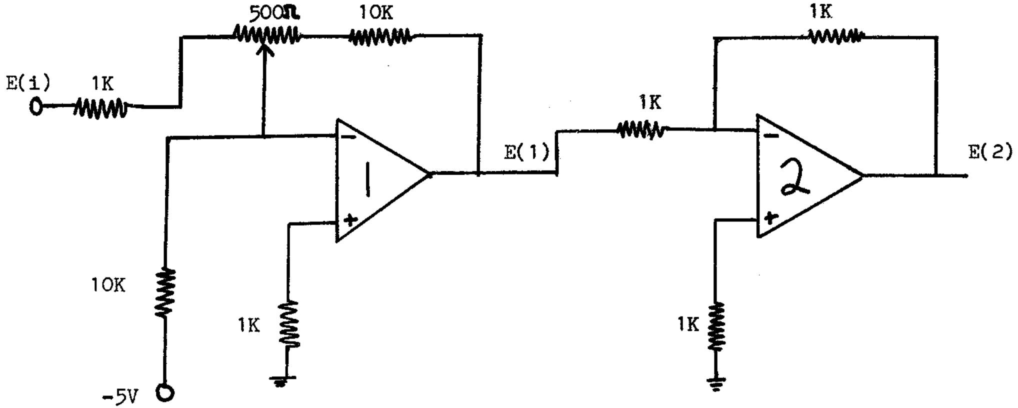
The second op. amp. in the circuit functions as an inverter, and it's output, E(2), depends on E(1) and the external resistances in the following way:

$$E(2) = -(E(1)R5/R4) \qquad (3)$$

Simplifying this equation, we obtain the following:

$$E(2) = -E(1) \qquad (4)$$

FIGURE 3 - HPLC Operational Amplifier



E(1) = -(10E(i) -- 5V)

E(2) = -E(1)

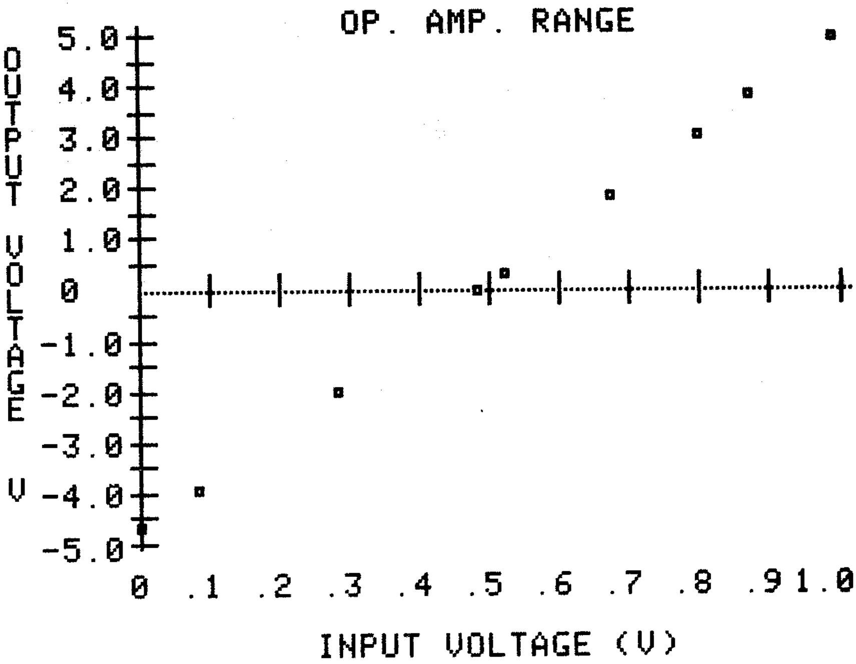
Using these equations, it is clear that an input voltage, E(i), at op. amp. 1 of 0V will produce an output voltage, E(2), at op. amp. 2 of -5V. Similarly, if E(i) equals +1V then E(2) will have a value of +5V.

In order to test the reliability of this op. amp., we simulated the input voltage from the VWD using the DAC. That is, using the DAC as a source of a variable DC voltage, we varied the input voltage, E(i), from 0 to 1V and recorded the resulting output voltage, E(2). A plot of Output Voltage vs. Simulated VWD Voltage shows the linearity of the signal modification by the op. amp.(Fig.4)

2) Fluorometer

As mentioned previously, the output analog signal from the fluorometer has a range that varies with different sensitivity settings on the fluorometer. To design an op. amp. for this instrument, it was therefore necessary to vary the resistance in the feedback loop of the amp. such that the ratio of R(f) to the input resistance offsets the change in the output signal of the fluorometer due to a change in sensitivity. An op. amp. that performs this conversion is shown in Figure 5. The equation that represents the behavior of this op. amp. is as follows:

Figure 4 - Linearity of HPLC Op. Amp.



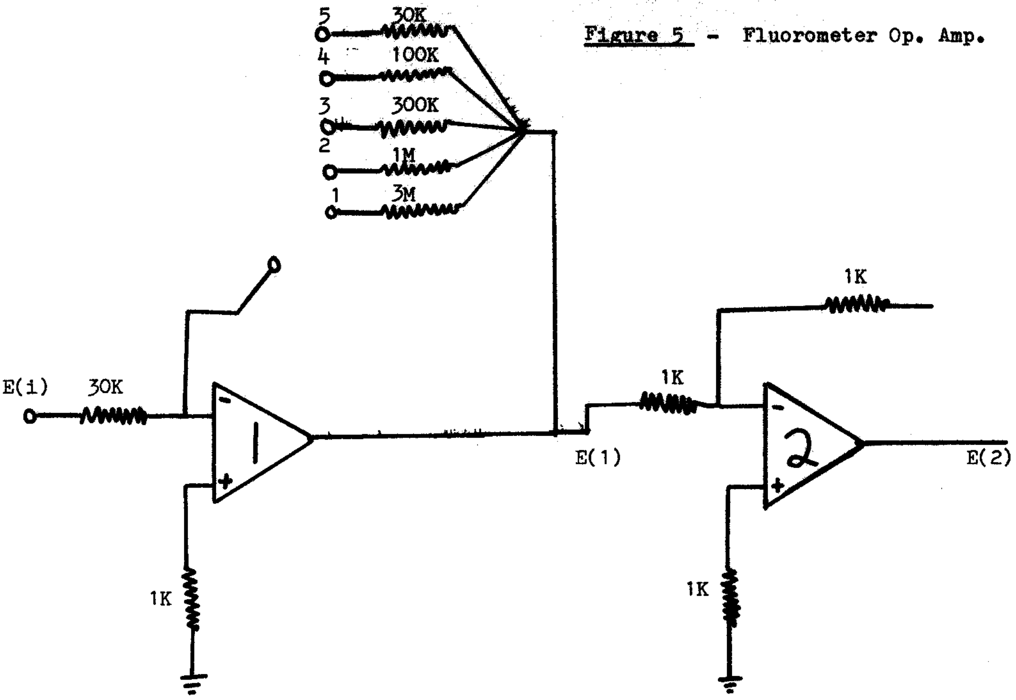
$$E(1) = -E(i)R(f)/30K \qquad (5)$$

It is important to note that, although there are actually six sensitivity settings on the spectrophotometer power supply, the op. amp. is designed to accomodate only the sensitity settings from one to five. The decision to exclude sensitivity six from our op. amp. design was based on the fact that this setting is never used.

It is necessary to adjust the Sensitivity and Zero settings on the fluorometer for each experiment. The user must connect a voltmeter to the output signal from the op. amp. and monitor this voltage in order to adjust the output range using these two settings. Ideally, the user will try to adjust the maximum emmission, at a particular wavelength, so that it corresponds to a voltage output of approximately 1V. Likewise, the baseline, which is obtained when the Filter setting is on S, should be adjusted so that its voltage is approximately zero volts. This will be more clearly explained in the user's manual.

Practically speaking, the output signal of the interface op. amp., after inversion, varies from 0 to 1V when the interface op. amp. sensitivity matches that of the fluorometer. With the signal confined to this range, the output of the interface op. amp. is fed into

Figure 5 - Fluorometer Op. Amp.



E(1) = -E(i)R(f)/30K

E(2) = -E(1)

the input of the HPLC op. amp. Recall that the HPLC op. amp. changes a 0 to 1V signal at its input to a -5V to +5V output signal compatible with the ADC.

Section II: SOFTWARE

This section describes the different computer programs necessary to process the discrete digital signal from the ADC. Since the programs are essentially the same for the fluorometer and the HPLC interface systems, we will present them only for the latter. Any important modifications made for the fluorometer interface system will be mentioned at the end of a particular program's description. It is important to note that most of the software, excluding the binary files used in the Interrupter program which were written in assembly language, was done in Apple BASIC. For a review of Apple BASIC the reader is referred to the Basic Programming Reference Manual. (3)

Programs will be analysed in the order that they would be used during an experiment. In addition, we will not go into a great deal of detail about the aesthetics of programming, i.e. how printouts were aligned and organized, nor will we attempt to analyse every step of each program. Rather, the function of each program, as well as the use of interesting Apple BASIC commands will be discussed.

1) Menu

The purpose of this program is to make the interface system accessible to the user. In short, the program allows the user to run one of six different programs that are stored on the interface diskette. The program prints a menu on the high resolution graphics screen of the Apple which presents the user with six different choices. This is accomplished using six "print" statements with the title of a particular choice in quotation marks following "print".(Table 2) The program communicates with the user in line 150 with an "input" statement that asks the user to select one of the choices by entering the number corresponding to a particular choice. In lines 180 - 207 the program makes use of the "On ...Gosub" statement which tells the Apple where in the program it should proceed when a choice is selected. In this case, the first five choices tell the computer to run a specific program stored on diskette. The sixth choice results simply in termination of the program.

The interesting aspect of this program is that it uses the CHR$ (4) command of Apple BASIC. This command allows instructions for the computer to be left within a program in the same format as these instructions would be entered by a user. In this program (line 250), note that the statement:

```
PRINT D$;"Run Interrupter"
                      where D$ = CHR$ (4) (line 20)
```

Table 2 - Menu Program

```
]LIST

10  REM  MENU FOR THE HPLC INTERFACE
20 D$ =  CHR$ (4): REM  CNTRL D
30  HOME
40  HTAB 5: PRINT "MENU FOR THE HPLC INTERFACE"
50  PRINT : PRINT
60  INVERSE
70  PRINT "1. DATA COLLECTION "
80  PRINT : PRINT
90  PRINT "2. PLOT "
100  PRINT : PRINT
110  PRINT "3. INTEGRATE "
120  PRINT : PRINT
130  PRINT "4. ADJUST BASELINE"
140  PRINT : PRINT
150  PRINT "5.LOAD DATA"
160  PRINT : PRINT : PRINT "6.QUIT"
170  NORMAL
180  PRINT : PRINT : PRINT
190  INPUT "ENTER A NUMBER AND PRESS RETURN. ";NB
200  PRINT
210  ON NB = 1 GOSUB 270
220  ON NB = 2 GOSUB 290
230  ON NB = 3 GOSUB 310
240  ON NB = 4 GOSUB 320
250  ON NB = 5 GOSUB 330
260  END
270  PRINT D$;"RUN INTERRUPTER
280  RETURN
290  PRINT D$;"RUN AXISLABEL"
300  RETURN
310  PRINT D$;"RUN ITGRT"
320  PRINT D$;"RUN BASESET"
330  PRINT D$;"RUN DATALOAD"
340  RETURN

]
```

instructs the Apple to run the program Interrupter. The Apple will run this program just as if the user himself had typed "Run Interrupter". This command will be extremely useful, for it allows us to communicate with the disk drives, i.e. to run other programs and load binary files. In fact, it is used in all of the programs of the interface system.

2) Interrupter

The key to the interface system is the Interrupter program. (Table 3) The function of this program is to monitor the discrete digital signal on one second intervals and store the digital value in successive memory locations in the Apple. This program can be divided into two distinct parts based on the computer language being used: an interrupt routine and a data acquisition and storage section. The interrupt part of the program is accomplished with three assembly language programs which are run within the Interrupter program. The data collection and storage segments of the program are written in Apple BASIC. These segments average the collected digital values and place the average values in successive memory locations.

After selection of Interrupter on the Menu, the computer proceeds to load three binary files into the Apple's memory.(lines 110-130) The interrupt part of

Table 3 - Interrupter Program

```
 LIST 0,400

10  HOME
20  GOSUB 630: REM  SETS OFFSET VALUE
70 CH = 4: REM  USING HARDWIRED CIRCUIT
80  HOME
90  HTAB 12: VTAB 15: PRINT "LOADING FILES...."
100 D$ =  CHR$ (4)
110  REM  THE NEXT THREE STEPS CONSIST OF LOADING THE MACHINE LANGUAGE PRO
     GRAMS NECESSARY FOR THE INTERRUPT PROGRAM
120  PRINT D$;"BLOAD A/DMOVE.OBJO": REM  LOADED AT 768
130  PRINT D$;"BLOAD INTRUPTOFF": REM  LOADED AT 857
140  PRINT D$;"BLOAD INTRUPT.OBJO": REM  LOADED AT 830
150 FLAG = 829: REM  MEM LOC FOR INTERRUPT FLAG
160 STRBS = 24576: REM  DATA STORED HERE & UP
170 ADDR = 49312 + CH: REM  SLOT 2 FOR A/D
180  REM  THE INTERRUPT ROUTINE IS AT LOC 768. IT STORES AN A/D VALUE AT 8
     28 AND SETS A FLAG AT 829.
190  HOME : HTAB 15: VTAB 15: FLASH : PRINT "INJECT SAMPLE": PRINT : PRINT
     : PRINT "PUSH RED BUTTON ON PADDLE 0": NORMAL
200  GOSUB 560
210 X =  PEEK (ADDR): REM  STARTS A/D
220  REM  A NULL READING OR PRIME FOR THE A/D
230 N = 0
240 NPT = 100: REM  # OF DATA PTS
250  POKE FLAG,0: REM  SETS FLAG AT 0
260  CALL 830: REM  TURNS ON 1 SEC INTRPTS
270  IF  PEEK (FLAG) = 0 THEN  GOTO 390
280 ANS = 0
290  REM  FOLLOWING LOOP IS AN AVERAGING LOOP FOR THE A/D
300  FOR I = 1 TO 20
310 ANS = ANS +  PEEK (ADDR)
320  NEXT I
330 X = ANS / 20
340  POKE STRBS + N,X
350 N = N + 1
370  POKE FLAG,0: REM   RESETS FLAG
380  IF  PEEK ( - 16384) > 127 THEN  GOTO 400: REM  CHECKS KEYBOARD
390  GOTO 270
400  POKE  - 16368,0


]
```

```
LIST 400,1000

400  POKE  - 16368,0
405 SEC = N
440  CALL 857: REM  TURNS OFF INTRPTS
450  HOME
451 MI =  INT (100 * N / 60 + .5) / 100
452  HTAB 5: VTAB 5: PRINT MI;" MINUTES HAVE ELAPSED."
460  HTAB 5: VTAB 10: INPUT "WOULD YOU LIKE TO SAVE THIS DATA?";Y$
470  IF Y$ = "YES" THEN 490
480  PRINT D$;"RUN MENU REVISED,D1"
490  INPUT "FILENAME?";F$
500  PRINT : INPUT "DISK DRIVE 1 OR 2?";NU
510  REM  SAVES DATA AS A BINARY FILE
520  PRINT D$;"BSAVE ";F$;", A24576, L";SEC;", D";NU
530  REM  CORRECT FORMAT WITH AN ADDED ;
540  GOTO 480
550  REM  TAKES YOU BACK TO THE MENU
560 X =  PEEK ( - 16287)
570  IF X > 127 THEN  GOTO 590
580  GOTO 560
590  POKE  - 16296,1
595  FOR I = 1 TO 5
600 C =  PEEK ( - 16336)
601  NEXT I
610  HOME : HTAB 12: VTAB 15: INVERSE : PRINT "COLLECTING DATA...": NORMAL

620  RETURN
630  REM  SETS OFFSET VOLTAGE AT - 5 VOLTS
640 SN = 2
650 VOLT = 8
660 BASE = 49280:CH = 3
670  FOR I = 1 TO 20
680 ADDR = BASE + SN * 16 + CH
690  POKE ADDR,VOLT
700  NEXT I
710  RETURN

]
```

this program is then begun by pressing the red button on game control #0.

Upon starting, the Apple sets the Flag value at 0 (Line 220) using the "Poke" statement. This is an important command in Apple BASIC which will be used later in this program for storage of the average digital values. This command stores a digital value, in this case 0, at a specified memory location. Here the memory location, 829, was set equal to Flag, so that the statement is written as follows:

```
POKE Flag,0
```

With the Flag set equal to zero, the program proceeds to line 230, which is where the binary files take control of the programming. The first in the series of assembly language programs is INTRUPT.OBJ0, which was loaded at memory location 830. (Note : the command in line 230 refers to memory location 830.) This routine sets up the interrupts by turning on the Apple clock board, line 034C-. (Table 4) It then stores the address of the beginning of the service routine, where the computer will proceed to when an interrupt occurs. The service routine referred to is actually the assembly language program A/DMOVE.OBJ0. (Table 5) The function, therefore, of the first binary file is to

Table 4 - INTRUPT.OBJO Program

```
*300L

0300-   A5 45       LDA   $45
0302-   48          PHA
0303-   8A          TXA
0304-   48          PHA
0305-   98          TYA
0306-   48          PHA
0307-   AD C7 C0    LDA   $C0C7
030A-   58          CLI
030B-   AD A1 C0    LDA   $C0A1
030E-   8D 3C 03    STA   $033C
0311-   EE 3D 03    INC   $033D
0314-   68          PLA
0315-   A8          TAY
0316-   68          PLA
0317-   AA          TAX
0318-   78          SEI
0319-   AD C8 C0    LDA   $C0C8
031C-   68          PLA
031D-   40          RTI
031E-   00          BRK
```

Table 5 - A/DMOVE.OBJO Program

```
*33EL

033E-   48          PHA
033F-   78          SEI
0340-   A9 00       LDA   #$00
0342-   8D FE 03    STA   $03FE
0345-   A9 03       LDA   #$03
0347-   8D FF 03    STA   $03FF
034A-   A9 01       LDA   #$01
034C-   8D C9 C0    STA   $C0C9
034F-   58          CLI
0350-   68          PLA
0351-   60          RTS
0352-   00          BRK
0353-   00          BRK
0354-   00          BRK
0355-   00          BRK
0356-   FF          ???
0357-   FF          ???
0358-   FF          ???
0359-   78          SEI
035A-   48          PHA
*
```

start the interrupts on the clock board and to tell the computer where to go when an interrupt occurs. In this case, the computer is instructed to run the service routine each time an interrupt occurs.

The service routine instructs the computer to accept interrupts from other peripherals. When an interrupt occurs, this routine reads the A/D value and stores the digital value in memory location 828 (line 030E-). Then, it stores a 1 in memory location 829, which sets the Flag. When the Flag is set at 1, which will be at one second intervals, the BASIC part of the Interrupter program:

1)looks at the A/D value (20 times).
2)calculates an average value.
3)stores the average value.
4)resets the flag to zero.

Although these steps will be explained in more detail later on, it is important to mention them here in order to understand the programming design.

The service routine will be repeated on one second intervals, and hence the A/D averaging and storing will be repeated on one second intervals.

When the Interrupter program reaches line 440, the computer is once again instructed to run an assembly language routine. This routine, stored at memory location 857, corresponds to the INTRUPTOFF binary file. (Table 6) The function of this routine is to turn off the clock board interrupts, lines 035B- and 035D, and to enable the system to be interrupted and have a response other than mentioned above. That is, when an interrupt occurs the computer will no longer run the service routine. Then, this routine returns the computer to the Interrupter program.

At this point, a further description of what the computer does when the Flag is set equal to 1 is in order. The first function performed within the BASIC part of the Interrupter program is to look at the A/D value using the Peek command. This command is the complement of the previously explained Poke command, in that it is used to look at the digital value stored in a particular memory location. The format used when looking at a chosen memory location is as follows:

```
X = PEEK(ADDR)
```

where ADDR stands for a particular address or memory location of interest.

When "peeking" at A/D values, the address is specified by an equation which is a function of the slot number for the A/D board and the channel number that is being

Table 6 - INTRUPTOFF Program

```
 359L

0359-   78          SEI
035A-   48          PHA
035B-   A9 00       LDA   #$00
035D-   8D C9 C0    STA   $C0C9
0360-   AD C7 C0    LDA   $C0C7
0363-   AD C8 C0    LDA   $C0C8
0366-   A9 18       LDA   #$18
0368-   85 23       STA   $23
036A-   58          CLI
036B-   68          PLA
036C-   60          RTS
036D-   78          SEI
036E-   60          RTS
036F-   58          CLI
0370-   60          RTS
0371-   03          ???
0372-   FF          ???
0373-   04          ???
0374-   05 06       ORA   $06
0376-   FF          ???
*
```

monitered.

ADDR = 49280 + (slot# * 16) + channel#

where

1)49280 is the starting address of all slot dependent locations.

2)slot# must be from 1 to 7.

3)channel# must be from 0 to 15.(4)

Since the slot number refers to the place where the A/D board is installed, in this case slot 2, this remains constant in the Interrupter program. Likewise, the channel (channel = 4 in the hardwired circuit) on which data are collected is always the same, so this is also a constant. It is important to mention that one can "peek" at memory locations other than those concerned with A/D conversion by simply specifying the address of interest.

We decided to have the program calculate an average (N=20) for the A/D value due to the fluctuations observed when monitering single values. It then stores the average digital value at memory locations from 24576 and up. It is worth noting that there are approximately 10,000 memory locations available for data storage. This means that, when sampling 60 average values per minute, this system is capable of storing almost 3 hours worth of data. This will be more than sufficient for applications involving the HPLC and fluorometer.

Lastly, the program resets the Flag value at 0, which allows the program to be interrupted at the next second interval by the assembly language programs.

Other peripheral functions of the Interrupter program are:

1)to generate the digital value for the D/A converter which is used as an offset voltage for the HPLC op. amp. (Subroutine 640) The program produces a voltage on channel 3 of -5V.

2)to calculate the elapsed time of data collection. (Lines 451-452)

3)to allow the user to save collected data to a data diskette. (Lines 490-520)

3) Plot

The next step is to display acquired data in graphical form. For the HPLC, we are interested in plotting average digital absorbance value vs. time. This will produce a chromatogram on the high resolution graphics screen of the Apple. On the other hand, for the fluorometer interface system the plot is of average digital emission value vs. wavelength, which yields a fluorescence spectrum. Although data are collected as a function of time for the fluorometer, it is possible to convert the time axis to wavelength by simply incorporating the scan speed for the emission monochromator into the plotting procedure. This will be explained in more detail following a discussion of the HPLC plotting program.

A) HPLC - Chromatograms

The plotting program (Table 7) for the HPLC interface system has four important functions. These are as follows:

(1)receives input from the user which specifies what range (in minutes) the program will scale and plot a chromatogram.

(2)finds the highest average digital value within a

<u>Table 7</u> - HPLC Plotting Program

```
 LIST 0,400

10 LE = PZ - PA: REM  CALC. LENGTH OF PLOT
20 TXTGEN% = 768: REM  SPEC. MEM. LOC.
30 D$ =  CHR$ (4): REM  D$ = CNTRL D
39  HOME
40  HTAB 5: VTAB 10: INPUT "PLOTTING RANGE (MIN.,MIN.)";PA,PZ
50  HTAB 10: VTAB 15: INPUT "TITLE ?";TITLE$
60  HGR2 : HCOLOR= 3: REM  HIGH RES SCREEN #2
70 LE = PZ - PA: REM  CALC. LENGTH OF PLOT
80  PRINT D$;"BLOAD TEXT"
90  CALL TXTGEN%
100 HOCH = LE * 60: REM  HOCH = RANGE(IN SECONDS)
110 MAX = 0: REM  SPECIFIES LOW VALUE THAT PEAK GREATER THAN
120  REM  THIS IS USED WHEN SCANNING FOR THE HIGHEST PEAK
130 TREPPEN = 279 / LE
140  REM  TREPPEN = STEPS(AUF DEUTSCH),DIVIDES OUR X AXIS INTO MINUTES
150  REM  PLOTS X AND Y AXIS
160  HPLOT 0,0 TO 0,147 TO 279,147
170  FOR I = 0 TO 150 STEP 7
180  FOR M = 1 TO 279 STEP TREPPEN
185  ONERR  GOTO 900
190  HPLOT M,I TO M + 1,I
200  NEXT M
210  NEXT I
220  FOR L = 0 TO 279 STEP TREPPEN
230  HPLOT L,157 TO L,147: REM  X HATCH MARKS
240  NEXT L
250  FOR N = 0 TO 279 STEP TREPPEN / 5: REM  DIVIDES MINUTES INTO 20/100 I
     NTERVALS
260  HPLOT N,152 TO N,147: REM  X HATCH MARKS
270  NEXT N
280 LA = PA
290  FOR HT = 1 TO 40 STEP 39 / LE
300  VTAB 21: HTAB HT: PRINT LA
310 LA = LA + 1
320  IF LA = PZ THEN HT = HT - 1
330  NEXT HT
340  HTAB 15: VTAB 23: PRINT "TIME(MINUTES)"
350  HTAB 15: VTAB 1: PRINT TITLE$
360  REM  NAME STARTING LOCATION
370 STRBS = 24576
380  REM  COMPENSATE FOR RANGE PROGRAM BY MOVING TO THE PROPER STARTING LO
     CATION
390 STRBS = STRBS + PA * 60
400  REM  FIND THE DIGITAL VALUE FOR THE LARGEST PEAK

]
```

```
LIST 400,1000

400  REM  FIND THE DIGITAL VALUE FOR THE LARGEST PEAK
410  FOR S = 1 TO HOCH
420 RES =  PEEK (STRBS)
430 STRBS = STRBS + 1
440  ON RES > MAX GOSUB 650
450  NEXT S
460 STRBS = 24576 + PA * 60
470  REM  PLOT THE GRAPH
480 RES =  PEEK (STRBS)
490 Y1 = 147 * (1 - (RES / MAX))
500 X1 = 0
510  FOR R = 1 TO HOCH
520 RES =  PEEK (STRBS + R)
530 Y2 = 147 * (1 - (RES / MAX))
540 X2 = R * 279 / HOCH
550  HPLOT X1,Y1 TO X2,Y2
560 X1 = X2:Y1 = Y2
570  NEXT R
580  HTAB 28: VTAB 22: PRINT "MAX = ";MAX
590  GET G$
600  HOME : TEXT
605  VTAB 12: HTAB 2: INPUT "WOULD YOU LIKE TO ADJUST THE RANGE?";Y$
606  ON Y$ = "YES" GOSUB 1000
609  HOME
610  VTAB 12: HTAB 8: INPUT "WOULD YOU LIKE A HARDCOPY?";Y$
620  ON Y$ = "YES" GOSUB 670
630  PRINT D$;"RUN MENU REVISED,D1"
640  END
650 MAX = RES
660  RETURN
670  PRINT D$;"RUN EPSON PLOT"
900  GOTO 590
1000  GOTO 10

]

]
```

selected range and scales that value to the full size of the high resolution graphics screen. In addition, it prints the maximum digital value, Max Value, on every chromatogram.

(3)draws an X and Y axis and plots the average digital value vs. time. This includes labeling of the X axis with hatch marks and a title. The user may also choose a title for the entire chromatogram.

(4)allows the user to make a hardcopy of a chromatogram by running the Epson Plot program.

The first of these functions is accomplished in line 40, where the user is asked to specify a plotting range, in minutes. This is extremely useful, since it allows enlargement of paricular regions of interest on a chromatogram by simply specifying the appropriate time range. In addition, this input is also used later on in the program to label the X axis in intervals of one minute. Samples of a full chromatogram and one in which a section of the chromatogram has been expanded are shown in figures 6,7, and 8. Given the range over which the program will function, the computer proceeds to scan that range for the maximum average digital value. In lines 400 to 450, the computer compares the average digital values with a low digit (Max = 0 (Line 110)) so that when a peak occurs, its maximum digital value will be greater than Max. When a peak occurs, its

Figure 6 - Plotting Sample of a Full Chromatogram

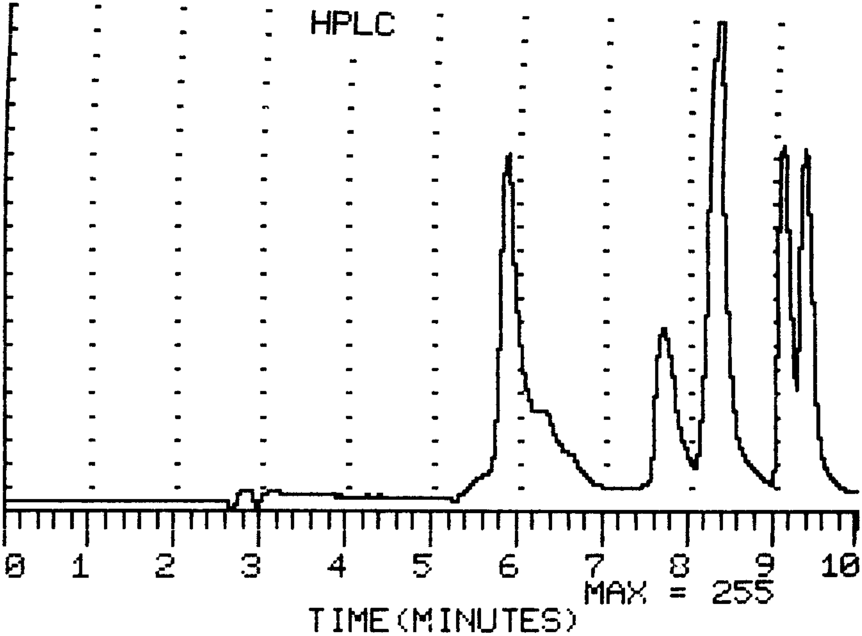


Figure 7 - Plotting Sample which Illustrates Expansion

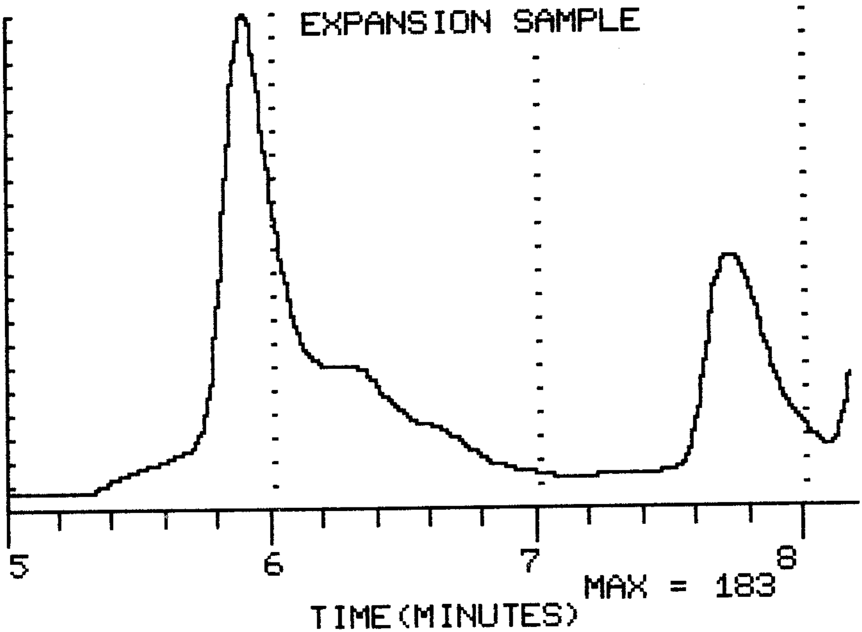
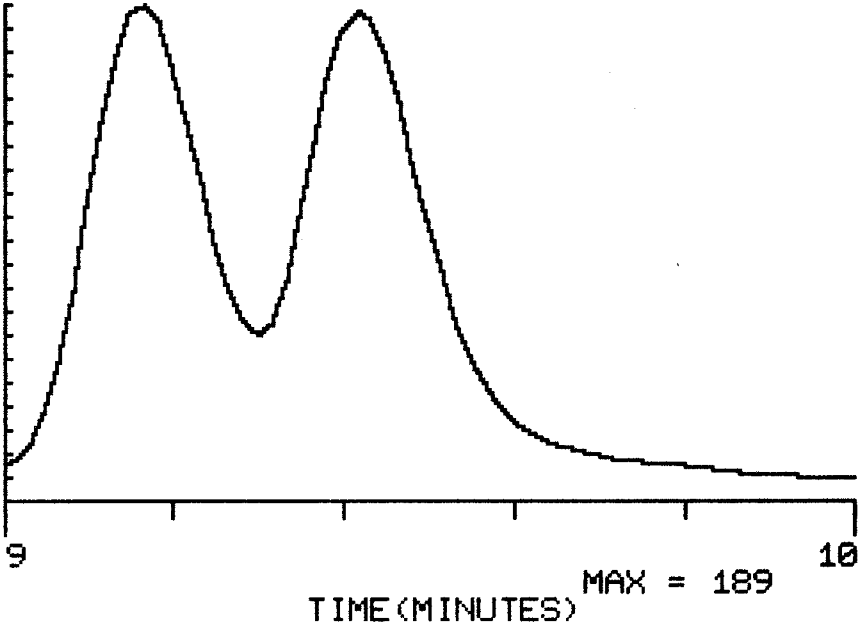


Figure 7 - Expansion Illustration



maximum digital value then becomes the new Max. value (line 440 + 650). The comparison continues until the Max value is the highest average digital value monitored. Scaling of the highest peak to the full size of the high resolution graphics screen is then accomplished in line 490, where the Max value is used to calculate the scaling factor.

Plotting of the X and Y axes is accomplished in lines 150 - 350. This section of the program plots hatch marks at minute intervals within a selected range and divides these intervals into fifths. This makes it easier to read a desired time from a chromatogram. In addition, this is the part of the program that labels the X axis using the entered range for the minute scale. The title is also printed at this point. The examples, shown previously, illustrate the format for these plots.

Finally, this program asks the user whether or not he would like to have a hardcopy of a chromatogram (Line 610). If the answer is yes, the computer then runs the Epson Plot program (Line 670) which prints, on the Epson dot matrix printer, whatever has been written on the Apple high resolution graphics screen. Since this program is a commercial one, we will not explain how the program works. A description of how to use the program is included in the HPLC Interface Operator's Manual, which is found in the appendix of this paper.

B) Fluorometer - Emission Spectrum

It is only necessary to make minor adjustments in the previous HPLC plotting program to plot an emission spectrum. Emission spectra are plots of average digital emission vs. wavelength; therefore, the major change involves relating the monochromator scan speed to the number of data points collected over an interval of time. The Medium scan speed, which is the only one we have chosen to program for, is 100 nanometers per minute. (5) In addition, recall that data points are collected on one second intervals, i.e. 60 data points per minute.

In line 40 (Table 8) the plotting range is entered in nanometers, and this wavelength range is then converted to a time range (in minutes) in line 72. Next the range (in seconds) is calculated in line 100. This conversion of ranges allows us to plot average digital emission values using the same routine as for the HPLC plotting program. (410-570) That is, the plot is actually average digital emission values vs. time. The only difference is that the nanometer scale is printed on the X-axis instead of the calculated time scale. In order to accomplish this, lines 130, 290, 310, and 340

Table 8 - FS Plotting Program

```
]LIST 0,400

5  REM  FLUOR PLOT
10 LE = PZ - PA: REM  CALC. LENGTH OF PLOT
20 TXTGEN% = 768: REM  SPEC. MEM. LOC.
30 D$ =  CHR$ (4): REM  D$ = CNTRL D
39  HOME
40  HTAB 5: VTAB 10: INPUT "PLOTTING RANGE (WAVE1,WAVE2)";PA,PZ
50  HTAB 10: VTAB 15: INPUT "TITLE ?";TITLE$
60  HGR2 : HCOLOR= 3: REM  HIGH RES SCREEN #2
70 LE = PZ - PA: REM  CALC. LENGTH OF PLOT
72 LL = LE / 100
80  PRINT D$;"BLOAD TEXT"
90  CALL TXTGEN%
100 HOCH = LL * 60: REM  HOCH = RANGE IN SECONDS
110 MAX = 0: REM  LOW VALUE THAT PEAK GREATER THAN
120  REM  THIS IS USED WHEN SCANNING FOR THE HIGHEST PEAK
130 TREPPEN = 2790 * 2 / LE
140  REM  TREPPEN = STEPS(AUF DEUTSCH),DIVIDES OUR X AXIS INTO MINUTES
150  REM  PLOTS X AND Y AXIS
160  HPLOT 0,0 TO 0,147 TO 279,147
170  FOR I = 0 TO 150 STEP 7
180  FOR M = 1 TO 279 STEP TREPPEN
185  ONERR  GOTO 900
190  HPLOT M,I TO M + 1,I
200  NEXT M
210  NEXT I
220  FOR L = 0 TO 279 STEP TREPPEN
230  HPLOT L,157 TO L,147: REM  X HATCH MARKS
240  NEXT L
250  FOR N = 0 TO 279 STEP TREPPEN / 5: REM  DIVIDES MINUTES INTO 20/100 I
     NTERVALS
260  HPLOT N,152 TO N,147: REM  X HATCH MARKS
270  NEXT N
280 LA = PA
290  FOR HT = 1 TO 40 STEP 780 / LE
300  VTAB 21: HTAB HT: PRINT LA
310 LA = LA + 10
320  IF LA = PZ THEN HT = HT - 2
330  NEXT HT
340  HTAB 15: VTAB 23: PRINT "WAVELENGTH (NM.)
350  HTAB 15: VTAB 1: PRINT TITLE$
400  REM  FIND THE DIGITAL VALUE FOR THE LARGEST PEAK

]
```

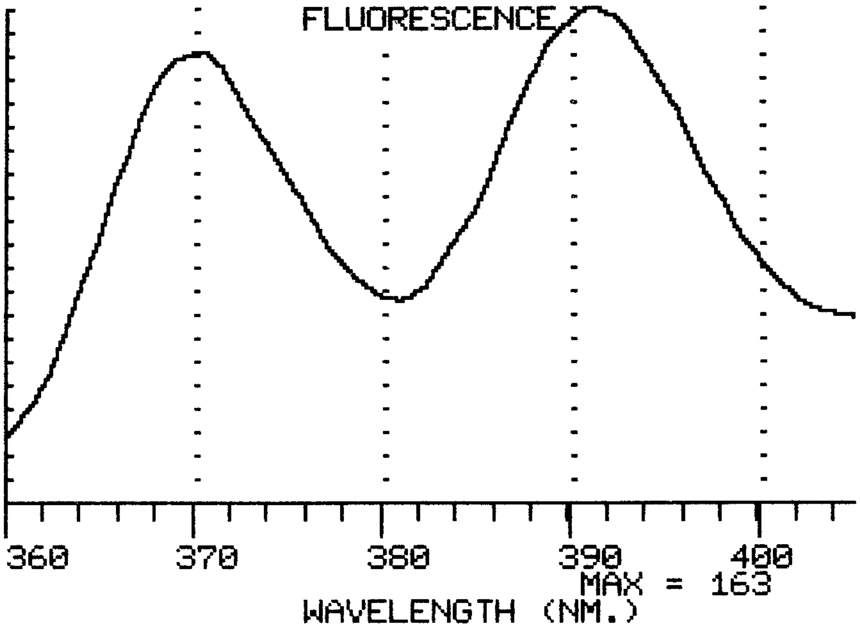
```
LIST 400,2000

400  REM  FIND THE DIGITAL VALUE FOR THE LARGEST PEAK
405 STRBS = 24576
410  FOR S = 1 TO HOCH
420 RES =  PEEK (STRBS)
430 STRBS = STRBS + 1
440  ON RES > MAX GOSUB 650
450  NEXT S
460 STRBS = 24576
470  REM  PLOT THE GRAPH
480 RES =  PEEK (STRBS)
490 Y1 = 147 * (1 - (RES / MAX))
500 X1 = 0
510  FOR R = 1 TO HOCH
520 RES =  PEEK (STRBS + R)
530 Y2 = 147 * (1 - (RES / MAX))
540 X2 = R * 279 / HOCH
550  HPLOT X1,Y1 TO X2,Y2
560 X1 = X2:Y1 = Y2
570  NEXT R
580  HTAB 28: VTAB 22: PRINT "MAX = ";MAX
590  GET G$
600  HOME : TEXT
610  VTAB 12: HTAB 8: INPUT "WOULD YOU LIKE A HARDCOPY?";Y$
620  ON Y$ = "YES" GOSUB 670
630  PRINT D$;"RUN MENU REVISED,D1"
640  END
650 MAX = RES
660  RETURN
670  PRINT D$;"RUN EPSON PLOT"
900  GOTO 590

]
```

were changed accordingly. An example of an emission spectra is shown in Figure 9.

Figure 9 - Plotting Sample of an Emission Spectrum



4) Integration

The next step is to calculate useful parameters for a particular plot. Since the integration program (Table 9) was specifically designed for the HPLC interface system, it will be explained with respect to chromatogramic data. This program calculates retention times, performs relative and absolute integration, and allows the user to make a hardcopy of these results. The discussion of this program will include a comparison of the relative integration method with the cut-and-weigh method.

This program has the following functions:

A)displays the previous plot to the user.

B)receives input which specifies the number of peaks to be integrated.

C)receives input which specifies the start, end, and selected baseline for a particular peak.

D)calculates retention times, as well as relative and absolute integration values.

E)allows the user to make a hardcopy of the integration results.

When a satisfactory plot has been obtained, the next step is to run the Integration program. Displaying

<u>Table 9</u> - Integration Program

```
 LIST 0,400

5 D$ =  CHR$ (4): REM  CNTRL D
10  REM  INTEGRATION PROGRAM
15 STRBS = 24576
20  DIM STPK(10): DIM BSPK(10): DIM AREA(10)
30  DIM NDPK(10): DIM PRCNT(10): DIM RTIME(10)
33 : PRINT AREA(L)
35 D$ =  CHR$ (4): REM  CNTR D
40  GOSUB 1000
42  GET G$
45  GOSUB 2000
50  HTAB 10: VTAB 16: INPUT "NUMBER OF PEAKS?";WW
62 N = 1
65  FOR I = 1 TO WW
70  PRINT "PEAK ";I: INPUT "START,END,BASE ?";STPK(N),NDPK(N),BSPK(N)
71 STPK = STPK * 60:NDPK = NDPK * 60:BSPK = BSPK * 60
72 STPK = STPK(N):NDPK = NDPK(N):BSPK = BSPK(N)
76 N = N + 1
77  NEXT I
80  IF STPK(N) < 0 GOTO 140
140 N = N - 1
150  FOR I = 1 TO N
160 WIDEPK = (NDPK(I) - STPK(I)) * 60
165 STRBS = 24576
170 BSAREA = WIDEPK * ( PEEK (STRBS + BSPK(I) * 60))
180 TAREA = 0
190  FOR TME = STPK(I) * 60 TO NDPK(I) * 60
200 TAREA = TAREA +  PEEK (STRBS + TME)
210  NEXT TME
220 AREA(I) = TAREA - BSAREA
230 TTAREA = TTAREA + AREA(I)
240  NEXT I
250  FOR J = 1 TO N
260 PRCNT(J) = 100 * AREA(J) / TTAREA
270  NEXT J
300  FOR K = 1 TO N
310 MAX = STPK(K) * 60
320 VM =  PEEK (STRBS + STPK(K) * 60)
330  FOR TM = STPK(K) * 60 TO (NDPK(K) * 60 - 1)
340 V1 =  PEEK (STRBS + TM)
360  IF V1 > VM THEN MAX = TM
365  IF V1 > VM THEN VM = V1
370  NEXT TM
380 RTIME(K) = MAX
390  NEXT K
400  GOSUB 750

]
```

```
 LIST 400,3000

400  GOSUB 750
405  PRINT : PRINT
425  PRINT
430  INPUT "WOULD YOU LIKE TO INTEGRATE AGAIN?";Y$
435  REM  STILL MUST REINIT DATA BEFORE USING
440  IF Y$ = "YES" THEN  GOTO 40
450  PRINT D$;"RUN MENU REVISED, D1"
750  HOME : HTAB 2: VTAB 12: INPUT "WOULD YOU LIKE A HARDCOPY?";Y$
755  ON Y$ = "YES" GOSUB 1500
800  HOME
802  PRINT "RET. TIME";: HTAB 13: PRINT "PEAK INTVL.";: HTAB 26: PRINT "%T
     OTAL";: HTAB 35: PRINT "ABS. AREA"
803  HTAB 2: PRINT "(MIN.)";: HTAB 17: PRINT "(MIN.)"
804  PRINT "**********   ***********   ******  ************"
805  FOR L = 1 TO N
806 RTIME(L) = RTIME(L) / 60
807 RTIME(L) =  INT (RTIME(L) * 10 + .5) / 10
809 N$ =  STR$ (PRCNT(L))
810  PRINT RTIME(L);: HTAB 13: PRINT STPK(L);"-";NDPK(L);: HTAB 26: PRINT
      LEFT$ (N$,5);: HTAB 36: PRINT AREA(L)
830  NEXT L
835  PRINT D$;"PR#0"
840  RETURN
1000  HOME : HTAB 15: VTAB 2: PRINT "INTEGRATION"
1001  HTAB 5: VTAB 5: PRINT "ENTER DATA AS FOLLOWS:"
1002  HTAB 7: VTAB 7: PRINT "START,END,BASELINE"
1005  HTAB 6: VTAB 10: PRINT "DON'T FORGET THE COMMAS OR <RETURN> WHEN ENT
     RY IS COMPLETED"
1006  HTAB 5: VTAB 12: PRINT "ENTER <ESC> TO DISPLAY LAST PLOT"
1100  RETURN
1500  PRINT D$;"PR#1"
1505  RETURN
2000  REM  DISPLAY GRAPHICS SCREEN
2009  POKE  - 16299,0
2010  POKE  - 16297,0
2015  POKE  - 16304,0
2020  GET G$
2030  TEXT : RETURN

]
```

this plot is accomplished in subroutine 2000. Using specific "Poke" commands, it is possible to have the computer display the last plot from the high resolution graphics screen. The addresses which are being "poked" and a discussion of their function can be found in the *Apple BASIC Reference Manual*. (6) This feature is convenient because it allows the user to identify locations of peaks during the actual integration program.

Next, the user enters the number of peaks to be integrated. (Line 50) This program is capable of performing integration of up to ten peaks. This upper limit is specified in lines 20-30, where input data as well as intermediate calculated values are dimensioned.

After viewing the plot and determining the Start, End, and Base time values of each peak, the user enters these values at line 70. The Base input may be different for each peak, in order to compensate for a drifting baseline.

At this point, a discussion of the method of integration performed by this program is in order. Basically, the method involves adding up all the average digital values within the chosen time interval (Lines 150-240) and subtracting the average digital value for the baseline (Line 220). This sum corresponds to the absolute integration value, and with it the relative percentages can be calculated (Lines 250 - 270). Then, the maximum value for each peak is calculated (Lines 300 - 390) The time at which a maximum value occurs corresponds to the retention time of that peak.

Lastly, the user is given the option to obtain a hardcopy of the integration results. If the printer is not connected to the system, the user should answer no here and integration results will simply appear on the Apple graphics screen. On the other hand, if the printer is connected to the system, a hardcopy of integration results may be obtained by answering yes to this question. Further instructions on the use of this program are included in the user's manual in the appendix of this paper.

As a test of the reliability of the integration method employed in this program, we compared the relative percentages for two peaks with the "cut-and-weigh" method (Table 10). The results are satisfactory, in that the percent deviation is less than 3%. The absolute integration values, which simply correspond to the digital sum, can be very useful when doing comparative concentration calculations.

Table 10 - Integration Comparison

```
]RUN
0
              INTEGRATION
    ENTER DATA AS FOLLOWS:
      START,END,BASELINE
     DON'T FORGET THE COMMAS OR <RETURN> WHEN ENTRY IS COMPLETED
    ENTER <ESC> TO DISPLAY LAST PLOT
         NUMBER OF PEAKS?2
PEAK 1
START,END,BASE ?5.3,7.0,5.3
PEAK 2
START,END,BASE ?7.5,8.15,7.5
 WOULD YOU LIKE A HARDCOPY?
                                 YES
RET. TIME    PEAK INTVL.   %TOTAL    ABS. AREA
 (MIN.)          (MIN.)
*********    **********    ******   ***********
5.9          5.3-7         75.97       4183
7.7          7.5-8.15      24.02       1323
```

Cut and Weigh

Ret. Time (min.)	Weight (g.)	% Total
5.9	.2351	74.27
7.7	.0605	25.73

5) Baseline Adjust

Sometimes during an experiment with the HPLC it is necessary to adjust the level of the baseline on the chart recorder. Likewise, certain applications with the interface system will require a range adjustment of the HPLC op. amp. For this reason, a program (Table 11) has been created which allows the user to monitor and alter the upper and lower limits of the HPLC op. amp. output voltage. These adjustments are accomplished using the 500 ohm variable resistor which was included in the HPLC op. amp. design.

This program "pokes" specific digital values into designated memory locations, which corresponds to having an offset voltage of -4.56V (Line 100 (V1)) and an input voltage, E(i), of 1V (Line 100 (V2)). Notice that channel three is being used for the offset voltage and channel 10 is being used for E(i). (Lines 30 + 70) Further, this program displays, on the graphics screen of the Apple, the output voltage, E(2), from the HPLC op. amp. as well as the corresponding digital value. With a small screwdriver, the user may adjust the lower limit by turning the screw on the 500 ohm variable resistor, which protudes from the box that encloses the HPLC op. amp. At the same time, the user can view the resulting changes on the Apple graphics screen. In

Table 11 -   Baseline Adjust Program

```
LIST

10  REM  CHECK OP.AMP.
20 ANS = 0:BASE = 49280
30 SN = 2:CH = 3:CB   10
40 V1 = 11:V2 = 153
50  HOME
60  GOSUB 140
70 ADDR = BASE + SN * 16 + CH: REM  ADDRESS FOR -5 VOLTS
80 ARDD = BASE + SN * 16 + CB
90  FOR I = 1 TO 20
100  POKE ADDR,V1: POKE ARDD,V2
110  NEXT I
120  GOSUB 180
130  END
140  HTAB 15: VTAB 2: PRINT "ADJUST BASELINE"
150  HTAB 15: VTAB 13: PRINT "DIGIT"
160  HTAB 30: VTAB 13: PRINT "VOLTAGE"
170  RETURN
180  FOR I = 1 TO 50
190 ANS = ANS +  PEEK (49316)
200  NEXT I
210 W = ANS / 50
220 ANS = 0
230 W =  INT (W * 100 + .5) / 100
240 Q = (W - 127.5) / 25.5
250 Q = ( INT (100 * Q + .5)) / 100
255  INVERSE : HTAB 15: VTAB 16: PRINT W: HTAB 33: VTAB 16: PRINT Q: NORMAL

265  IF W > 100 THEN  GOSUB 500
266  IF W < 10 THEN  GOSUB 520
270 KEY =  PEEK ( - 16384)
280  IF KEY > 127 THEN 320
290  NORMAL
295  HTAB 10: VTAB 19: PRINT "HIT ANY KEY TO CONTINUE"
300  GOTO 180
310  RETURN
320  POKE  - 16368,0
330  NORMAL
340  HTAB 5: VTAB 20: PRINT "DONE"
350  GOSUB 370
355  PRINT  CHR$ (4);"RUN MENU REVISED"
360  END
370  FOR I = 1 TO 10
380  POKE ARDD,128
390  NEXT I
400  PRINT "VALUE RESET"
410  RETURN
500  HTAB 18: VTAB 12: PRINT "UPPER LIMIT"
510  RETURN
520  HTAB 18: VTAB 12: PRINT "LOWER LIMIT"
525  NORMAL : HTAB 19: VTAB 16: PRINT "
530  RETURN

]
```

effect, the Apple is being used as a digital voltmeter.

When the adjustment of the lower limit has been completed, the user can flip a switch on the HPLC op. amp.(from Standard to Baseset) that will change the input at E(i) from 0V to 1V. This way, it is possible to see the upper limit of the new range.

Further intstructions on the use of this program are included in the user's manual in the appendix of this paper. One should note that baseline adjustment will not be necessary in most cases.

6) Load Data Files

The last program on the Menu allows the user to load previously obtained data into the memory of the Apple. The program (Table 12) was designed so that the Interface diskette must be in diskdrive one and the Data diskette must be in diskdrive two. With that, the program asks the user whether or not he would like a display of the available data files in line 100. If the user answers yes, then a "catalog" is displayed on the Apple's screen. (Line 152) Then, the user is asked to enter the name of the data file he wishes to load. (Line 25) If the file is successfully loaded, the program returns the user to the Menu. (Line 35) If not, the user may try to load another file or correct any mistake that might have been made in the entry.

Table 12 - Load Data File Program

```
]
 LIST

5 D$ =  CHR$ (4): REM  CNTRL D
10  REM  LOAD DATA FILE
11  HOME
12  GOSUB 100
15  HOME : PRINT : HTAB 5: PRINT "YOU CAN LOAD PREVIOUS DATA INTO"
20  PRINT : HTAB 3: PRINT "MEMORY BY SIMPLY SPECIFYING THE"
22  PRINT : HTAB 3: PRINT "FILENAME."
25  PRINT : INPUT "FILENAME?";FI$
27 N = 2
29  ONERR  GOTO 39
30  PRINT D$;"BLOAD ";FI$;", D";N
35  PRINT D$;"RUN MENU REVISED,D1"
39  HOME
40  HTAB 12: VTAB 14: PRINT "NO FILES FOUND"
42  HTAB 12: VTAB 16: INPUT "TRY AGAIN?";Y$
44  IF Y$ = "YES" THEN 15
45  IF Y$ = "NO" THEN 35
100  HTAB 5: VTAB 2: INPUT "CATALOG?";Y$
110  IF Y$ = "YES" THEN 150
120  RETURN
150  HOME : INVERSE : PRINT "AVAILABE FILES": NORMAL
152  PRINT D$;"CATALOG,D2"
154  INVERSE : PRINT "HIT ANY KEY TO CONTINUE": NORMAL : GET G$
156  GOTO 25

]
```

(Line 42)

This program facilitates quick processing of data which has been stored on diskette. In fact, most of the time processing of the data will occur after the actual run has taken place.

Although we are quite satisfied with the results of this project thusfar, there is still room for improvement in certain areas. Since we now realize the potential of this system, it is appropriate to mention our future research goals.

First, the method of starting data collection on the HPLC should be changed. It is possible to use the Event option on the HPLC to program a contact closure to occur when the Start button is pressed on the HPLC. This contact closure, or short circuit, could then be monitored by the A/D convertor using software similar to that used for data collection. In short, pressing the Start button on the HPLC would trigger the start of data collection.

There are some minor alterations of the plotting program which should be made. For example, it would be more convenient if plot titles could be typed directly on the high resolution graphics screen after a plot has been displayed. Once the title is specified, the user should then be able to move it to a desired location on the screen using designated keys on the Apple's keyboard.

It would also be useful to perform spectral correction for fluorescence emission and excitation spectra. For emission spectra the non-linear correction curve corrects for variation in detector sensitivity and emission monochrometer efficiency as a function of wavelength. On the other hand, the correction curve for excitation spectra corrects for variation of lamp intensity and excitation monochrometer efficiency as a function of wavelength. Correction factors are then divided into actual emission and excitation readings in order to obtain the true emission and excitation values. We plan to plot these correction curves and calculate correction factors for the appropriate nanometer readings, i.e. on one second intervals. Then, the correction values will be included in a DATA statement in the fluorometer's plotting program. This way the user will be able to print both the actual and corrected fluorescence spectrum.

Next, an optional automatic integration program would be very useful. This would allow the user to obtain relative and absolute integration results without making an entry. In other words, the computer would scan the average digital values for a plot and, using a slope calculating routine, define peak intervals automatically. This would not replace the present integration program; rather, it would be available for those who did not wish define peak intervals themselves. In addition, it might be easier if the present input format for the integration routine were changed. It has been suggested that entering peak interval data while viewing the plot would be more convenient.

Future research also includes designing interface systems for other chemical instrumentation. For example, we are considering a flash photolysis interface, which would involve monitering the A/D signal on a much smaller time interval.

Finally, the interface design research has been written up on a word processor. This allows us to make corrections or additions to the user's manual easily.

REFERENCES

(1) A/D + D/A Operating Manual, Mountain Computer Inc., 1980

(2) IBID, Chapter 3

(3) BASIC Programming Reference Manual, Apple Computer Inc., 1978

(4) A/D + D/A Operating Manual, Mountain Computer Inc., 1980, pg. 8-2

(5) Instructions Manual Spectrophotofluorimeter, Perking Elmer Norwalk, Conn., pg. 2-11

(6) BASIC Programming Reference Manual, Apple Computer Inc., 1978

APPENDIX

The purpose of this manual is to describe the use of the HPLC Apple Two+ system. In order to accomplish this, a standard procedure has been created, whereby the user plays a relatively small role in specifying certain parameters of the system. Of course, it is also possible for the user to fit this system to a particular application, in which case a more in depth understanding of the function and potential of this system is necessary.

SETUP

To begin, one must move the computer to the instrumentation lab and make the proper external connections. The latter part of this task should be relatively straightforward, since the electrical connections will involve simple connections with banana plugs. This will be more clearly explained and diagramed in the following section. Note: The Apple is very sensitive to being handled and extreme care must be taken during this portion of the interfacing procedure. For this reason, you should closely follow these steps:

1)Turn off the switch on the main power outlet and then unplug the main power plug.

2)Unplug the ribbon cord (grey with red and green) which connects the printer to the Apple. To do this, gently fold the two metal clips, located on opposite sides of the plug in the left rear of the printer, toward the printer. This will release the plug so that it can easily be removed and thereby freeing the Apple from the printer. Keep this ribbon cable with the Apple and avoid entangling it with any other wires from the printer.

3)Remove all diskettes from the disk drives, since slight movement of the drives may scratch any disks left there.

4)With a partner, move the Apple onto the computer cart taking care not to disrupt any of the connections in the rear of the Apple. Make sure that there is a screen, two diskdrives, two blue ribbon cables, one grey ribbon cable (from the printer), two paddle game controls, and an extension outlet attached to the apple. If there is any doubt at this point, please don't hesitate to ask Prof. Werner or Prof. Hull.

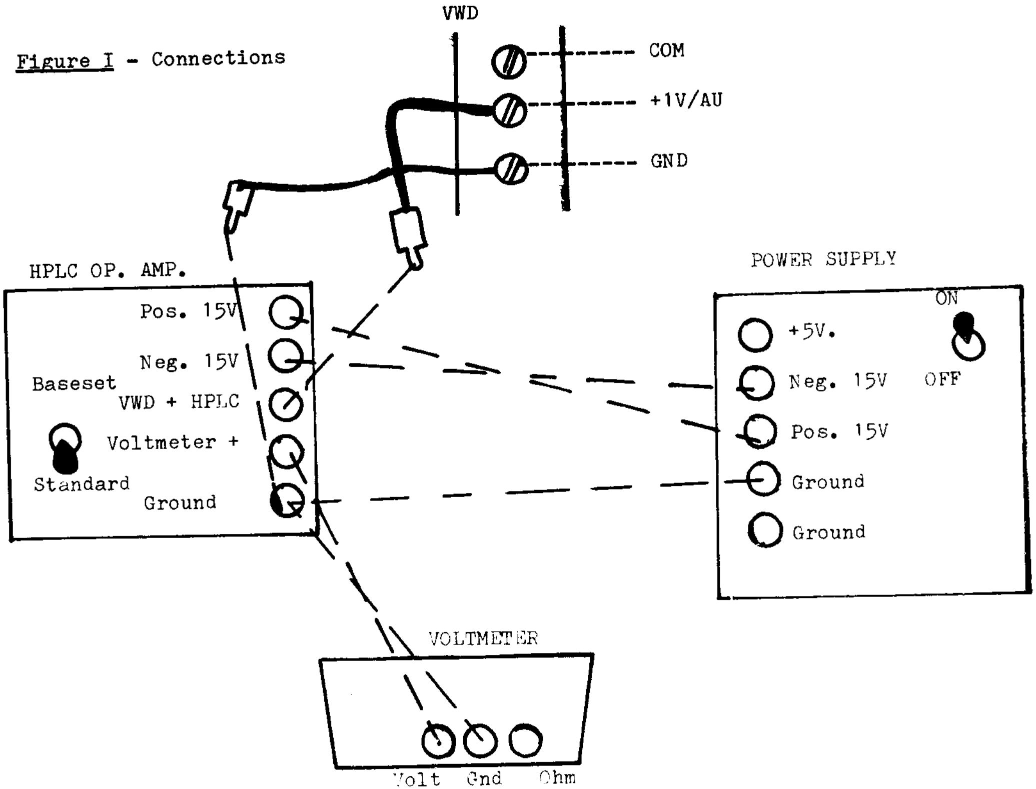
With the computer securely on the cart, wheel the cart to the instrumentation lab. Bring the cart behind the bench where the HPLC is located. Plug in the main power plug, but don't turn the min power switch back on yet!

Once you start moving things around, it is your resposibility to make sure everything is returned to it's original location. If any problems occur during the run, first note them in the interface logbook and then report them to a member of the chemistry department. In any case, sign your name in the logbook when the computer has been returned to the computer room.

CONNECTIONS

As mentioned previously, this portion of the interface should be relatively simple, since all of the electrical connections have been clearly labeled and diagrammed (Figure I). Match the labels from the operational amplifier (op. amp.) with those on the power supply, which you will find in the computer room with the interface diskettes on the top shelf of the white cabinet. You will notice two inputs on the op. amp., HPLC + VWD (variable wavelength detector) and VOLTMETER +, which don't have a match on the power supply. The voltmeter input must be connected with the positive terminal of a voltmeter by a banana cord. This is essential, so that the electrical signal doesn't drift, and, in addition, it allows the user to moniter the output of the op. amp. This will be useful when adjusting the baseline, which is discussed in a later

Figure I - Connections



section. The HPLC input on the op. amp. should have the output from the VWD plugged into it. This is the signal being monitored, so be sure not to forget this connection. Of the two outputs from the VWD, connect the one with the white label to the HPLC input and the other (yellow) banana plug to ground. Finally, be certain that the switch on the op. amp. is on standard.

(Note: At this point nothing plugged into an electrical outlet should be turned on. Instructions for this come in step one of the software section.)

SOFTWARE

At this point, we are almost ready to start taking data from the HPLC. The only element missing is the software which runs the system. You will find the two disks located in a black box labeled HPLC INTERFACE on the top shelf of the white cabinet in the computer room. When finished, please return the box to the location to make it easier for others to find them. This box contains the HPLC INTERFACE and DATA STORAGE disks. Beginning with data collection, the following steps will work through a standard run.

Data Collection

Step 1

Gently slide the HPLC INTERFACE disk into disk drive number one and the DATA STORAGE disk into disk drive number two, then close the black lids securing both of the disks. Don't force the disks into the drive, because the disks, as well as the drives, are very sensitive. Even a slight scratch on one of the disks might ruin the programs or data stored there. Next turn on the power by switching on the extension outlet. The red light on drive one will go on, and the disk will spin for about 15 seconds. During this time make sure that the power supply is plugged into an electrical outlet, and then turn on the power supply. When the red light goes off and the whirring noise has ceased, a menu will appear on the screen.

Step 2

The menu on the screen presents six different choices. The command on the screen says to select a number and press (Return).(Note: Commands within brackets refer to keys on the keyboard) For the standard run, the user should select number one, which corresponds to the interrupter program. This selection causes the computer to load the necessary binary files for the interface.

Step 3

Once the files have been loaded, the Apple flashes the following message on the screen:

Inject Sample

This signifies that the Apple is ready to start taking data. To inititate data collection the user must 1)inject the sample, 2)push the start button on the HPLC, 3)push the red button on game control 2, and 4)turn on the recorder at approximately the same time. Naturally, this is difficult even for the most gifted, since it involves being in four places at one time. For now this will have to do, and it's suggested that this is done with a partner. However, future plans include reducing the number of tasks to make it easier for the user.

Step 4

To terminate data collection, press the space key. A question will be posed as to whether or not the data should be forgotten. Most of the time the data should be saved to disk; therefore, answer no to this question. If it would be more appropriate to view the chromatogram before making this decision, answer yes and proceed to the plotting program. There will still be another chance, called Datasave, to save these data to disk.

When saving data it is necessary to specify a filename. This is of course a personal choice, except that choosing a name that's already been used on the data disk should be avoided; otherwise, the older version of the file will be deleted. For this reason, before entering a filename take note of the existing filenames as they are displayed on the screen. It is advisable to choose a filename that is pertinent to the nature of the data being saved, or else it might be difficult to know which filename corresponds to a certain run.

Display Data

Once the data have been saved to disk, the user will be returned to the menu. The next logical step is to plot the data, which corresponds to choice two on the menu. With this selection, the computer will ask for a range over which it should scale and plot the data stored in memory. To start, this entry should be the range of the entire chromatogram. For example, if the run were seven minutes long the entry would be:

```
0,7 (Return)
```

Next, the user has the option of selecting a title. If the user would rather not title a plot, he should simply press (Return) in response to this question. For instance, one might choose the following title:

```
HPLC RUN 1  (Return)
```

There is no need to capitalize, for all letters will be printed in capital letters on the plot. Once the chromatogram has been displayed in its entirety, pressing the space bar will take the computer out of the graphics mode. It will then be possible to make a hardcopy of the display, which is explained in the following section, if the Apple is connected to the printer. It should be noted that it is possible to blow up a particular region of a plot using this program. That is, the computer can scale an entered range to the full size of the graphics screen. For example, if the portion of interest is between five and seven minutes, one would simply enter that range using whole numbers as follows:

5,7 (Return)

Samples of a total chromatogram and a "rangeplot" are found on page .

Make a Hardcopy

At this point it may be useful to obtain a printout of the chromatogram. The first requirement for this program is that the Apple be connected to the printer. To make a hardcopy, answer yes to the last question at the end of the display data program, and the computer will load the necesary files. A new display will appear on the screen, which has a different format then the other programs thusfar described. The difference is this display has default values for the first four choices on the screen. The values, which are printed inversely, i.e. black on white, are the default values, and to change these values press the number corresponding to that line. The inverse type will move over to the next value to show that the change has been accomplished. In order to obtain a printout, it is essential that the first default value, which refers to the choice of pages (1 or 2) on the high resolution graphics screen, be changed from 1 to 2. To do this, simply press one. Otherwise, selections are personal preferences, so that the user decides whether or not to have a normal or enlarged printout. Likewise, the centering of the plot, as well as the mode of the printing, are selections made by the user. In order not to damage the printer, refrain from using the printer in the inverse mode for an extended period of time.

Once the parameters have been specified, check to see what is about to be printed by selecting number six. Before starting to print, be certain that the printing head is at the desired position on the page. This is controlled by two white buttons, labeled FF and LF, located beneath the ON LINE button on the printer. To move the printer head to the beginning of a new page the printer must be off line, which means that the green light beside the ON LINE button is not illuminated. With that, the page can be moved using the LF (line feed) button. When the page is in the desired position switch the computer back "on line" and select number seven of the plotting program. To print the enlarged version, it will take from two to three minutes. Once the printer has stopped, and the display consists of the various default values, (Cntrl C) will return the computer to the menu.

INTEGRATION

Once the data have been plotted, the user can perform relative and absolute integration and calculate retention times of selected peaks. This corresponds to the "integration" choice on the menu.

First, this program will display directions on the Apple graphics screen, which explain the input format for the integration program. When the user hits the (ESC) key, the present chromatogram will be displayed on the Apple high resolution graphics screen. At this point, the user must note the following:

1)the number of peaks to be integrated.

2)the "start" and "end" of each peak to be integrated.

3)the selected baseline for each peak.

It is advisable to write down these parameters while viewing the chromatogram.

With that, pressing any key will return the user to the original screen to enter the appropriate information. First, the number of peaks will be entered. (Note: this program is capable of integrating up to ten peaks.) Next, the peak intervals, i.e. Start, End, Base, are entered for each peak. After the peak interval for the last peak has been entered, the user may choose to obtain a hardcopy of the integration results. As with the plotting programs, it is essential that the Apple be connected to the Epson printer in order to obtain a printout of these results. It is also possible to have results displayed on the Apple graphics screen. The latter will be useful when integration is performed in the laboratory. A sample of the input format and integration results display is shown in Table 10. A comparison of the Apple's integration method with the cut-and-weigh method is also found in this table.

LOAD DATA FILES

Most of the time, data will be processed after the experiment is completed. This means that data files must be retrieved from disk and stored in the Apple's memory, in order to process data. Selection five on the menu allows the user to look at the available data files on a Data Diskette and load a particular file into the Apple's memory.

It is essential that data diskettes always be inserted in diskdrive 2 and that the interface diskette be inserted in diskdrive 1.

To see a display of the available data files, the user should answer yes to the initial question of this program. On the other hand, if the user is certain of the title of a particular data file, he may answer no here, and simply input the filename. Either way, after the data file has been loaded, the user will be returned to the menu. In the event that an error was made in entering the title of the data file, the user will be asked whether or not he wishes to try to load that file again.

MAKE A DATA DISKETTE

A data diskette is simply an initialized diskette. To initialize a diskette it is necessary to have a greeting program. If you would like to write this program yourself, it is clearly expained on page 14 of The DOS Manual. Otherwise, the following instructions will also enable you to initialize a diskette.

1)choose selection [illegible] on the menu.
2)type "New". (Return)
3)type "Load Ciao". (Return)
4)remove the interface diskette from diskdrive 1!
5)place the new diskette in diskdrive 1.
6)type "Init ciao". (Return)

The diskdrive will spin and make noises for about a minute. When the light on diskdrive one goes off the initialization is finsihed. Then, simply label this disk appropriately.

ADJUST BASELINE

It should be emphasized that it is not necessary to use this program for most runs on the HPLC. However, if a situation arises where the range of the HPLC op. amp. must be adjusted, this program will help perform that adjustment.

First, it is necessary to connect the power supply and voltmeter to the HPLC op. amp. as shown in Fig. I. The only difference is that the VWD is not connected to the op. amp. Note that the A/D and D/A cables must also be connected.

To adjust the lower limit of the op. amp. , i.e. the baseline, the toggle switch on the HPLC op. amp. should be set on standard. When everything is set and plugged in, the user makes selection four on the menu. A digital voltmeter will be displayed on the right side of the screen which will show the lower limit of the HPLC op. amp. output, E(2). This may be adjusted using the variable resistor which protrudes from the HPLC op. amp. This reading corresponds to the application of zero volts to the op. amp's signal input, E(i), with a -4.74V offset voltage. Remember that the A/D convertor

UN82 MEYER, J.R. MICROCOMPUTER INTERFACING WITH, ETC.
M612m/1982 CHEMISTRY HRS. 5/82 SHT. 2 OF 2

END

will not accept voltages less than -5V.

To adjust the upper limit of the op. amp's. range, simply switch the toggle on the HPLC op. amp. to the <u>baseset</u> setting. With that, adjustments can be made using the variable resistor. This corresponds to applying 1V to the op. amp's. signal input and with a -4.74V offset voltage. Remember that the A/D convertor will not accept voltages greater than +5V.

If you decide to adjust the baseline, please do not forget to reset the variable resistor at the end of your experiment. The lower limit should be -4.76V and the upper limit should be +4.72V.

For more involved users, another way to change the range of the HPLC op. amp. involves changing the offset voltage within the baseline adjust program. This corresponds to changing the digital value of V1 in line 40 of this program. (Table 11) When a satisfactory range is obtained, the same change must be made in the Interrupter program. (Line 650, Table 3).