PUMPING SYSTEM MONITOR OPERATIONS MANUAL

Prepared by

C.E. Brockway, P.E.
C.W. Robison, P.E.
S. Kolar

University of Idaho

Water Resources Research Institute

June 1992

TABLE OF CONTENTS

INTRODUCTION 1

SETUP 2

Hardware Description and Checklist 4

Tools and Supplies 4

Hardware Installation 5

Components 5

Sensors 5

PROGRAM CHECKLIST AND DESCRIPTION 9

PROGRAM STORAGE AND LOADING 10

Storage (RAM Disk) 10

Loading (Calculator) 14

OPERATION 15

Start–up (Initialization) 15

Operational Checks 18

Power–up 18

Information Display 18

Read and Write Pointers 19

Breaking In 19

Departure 20

DATA TRANSFER TO THE RAM DISK 21

DATA TRANSFER FROM RAM TO PC 22

APPENDIX 1 23

APPENDIX 2 25

APPENDIX 3 43

LIST OF FIGURES

Figure 1. 3

Figure 2. 7

Figure 3. 12

PUMP STATION MONITOR
OPERATIONS MANUAL

INTRODUCTION

The primary purpose of a pump station monitor is to provide the operator with real time information regarding the performance of the pump station and to record the performance parameters for later analysis. The performance of a pump station is determined from four basic components: input power, flow rate, discharge pressure, and input pressure. A pump station monitor requires a pump station to be equipped with pulse emitting watt meter(s), flow meter(s), and pressure transducer(s). This instrumentation allows the monitor to determine input power, flow rate, discharge pressure, and input pressure. Given these parameters, the calculator is programmed to provide the station efficiency, accumulated flow, and a number of other parameters. This data is collected and recorded on a real-time basis, and thus is "Time Tagged".

The heart of the monitoring system is the HP-41CX calculator. It performs the calculations, controls the scanning process of the sensors, conversion of the sensed parameter into standard units, and stores the results for later retrieval. The remaining components of the monitoring system, excluding the sensor portions, are a digital data acquisition interface (CMT-200) and the UI-MAD analog to digital conversion multiplier unit.

While an extensive knowledge of the HP 41 series calculators is not necessary, you must be familiar with the basic operation of the calculator, the XYZT stack, register allocations, the ALPHA keyboard, and program execution. This manual is not meant to replace the HP-41CX, HP-IL loop, CMT, and HP thermal printer manuals; further information and procedures can be found in these manuals.

The computational memory area of the calculator consists of 5 memory registers. These registers are really just storage bins for storing numbers which the calculator can

perform arithmetic operations on. Each bin, or register, has a name, and the registers can be thought to be arranged as shown below:

```
REGISTER NAME
      T            LAST X
      Z
      Y
      X
```

Notice that the registers X thru T are shown one above the other. This is referred to as the "register stack".

Throughout this manual, HP41 keystrokes and commands will be indicated with the use of brackets, braces and single quotes. Square brackets indicate a key should be pressed while the calculator is not in monitoring mode. Curly braces {} indicate the characters should be entered using the blue lettered keys. Single quotes indicate the blue lettered key should be pressed while the calculator is in monitoring mode.

SETUP

The diagram in figure 1 shows a sketch of a generic pump site, with typical locations for sensors and monitoring equipment. A judicious choice for locations of monitoring equipment as near as possible to the pump will make the initial setup of the air tube and sensor lines easier and less expensive. Here are a few pointers:

Calculation of the energy used and system efficiency is based on the application of the Bernoulli equation between the water level in the well or stream and the discharge into the distribution system. The placement of the sensors defines the type of efficiency being monitored. Also, the sensor positions describe how you are implementing the Bernoulli equation.

The location of the low pressure sensor is generally fixed by the location of the installed air tube.

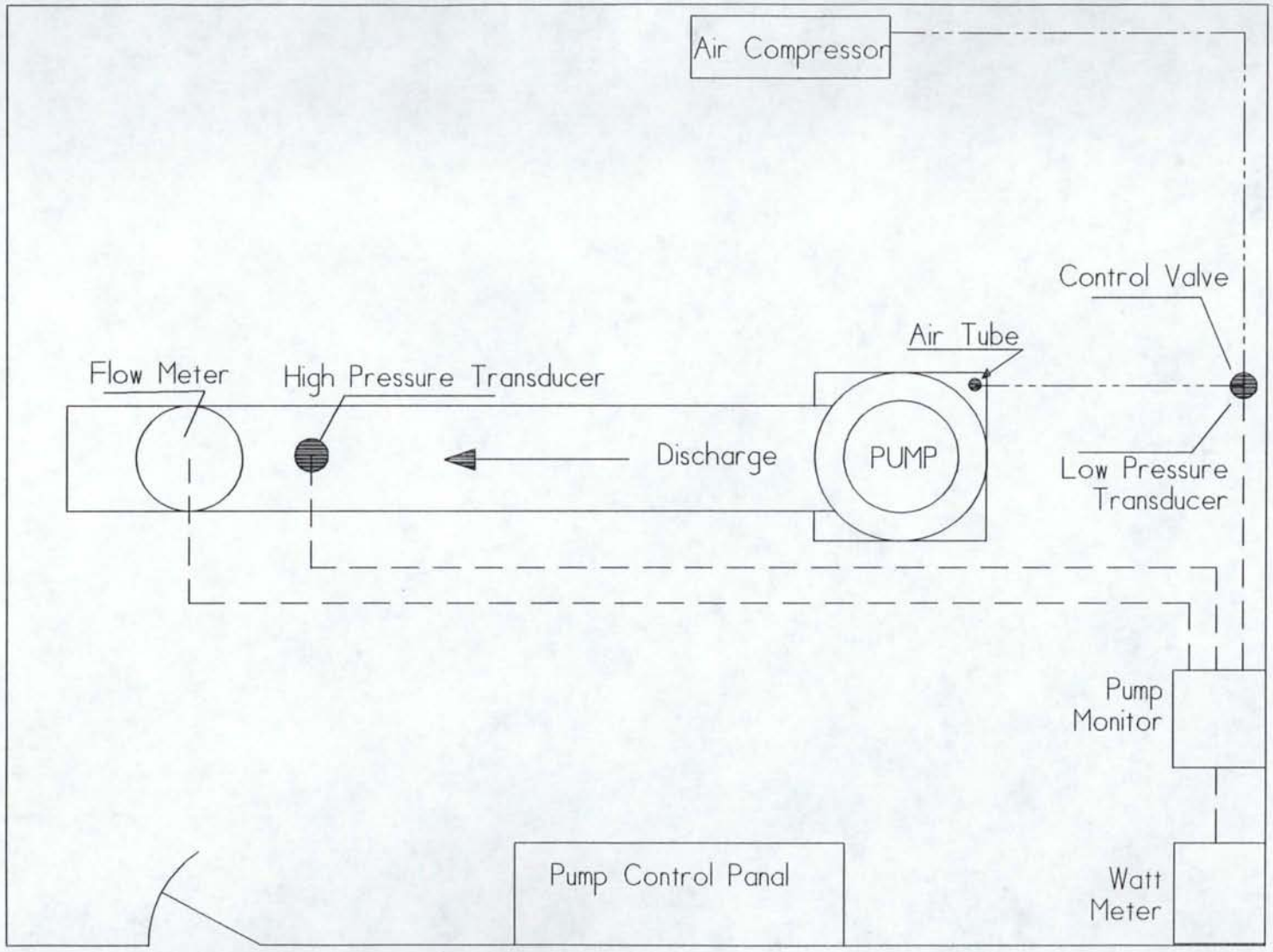

Figure 1. Typical Pump Station Monitor Installation

The location of the high pressure tap should be as near as possible to the pump on the discharge end. However, it should be in a section of the pipe which is free of air entrapment to give an accurate pressure reading.

Select the placement of the monitor box in as central a location as possible with respect to the different sensors. Sensor leads should be placed out of the way of traffic and should not interfere with operations.

HARDWARE DESCRIPTION AND CHECKLIST

You will need all of the following pieces of equipment and hardware for the site setup.

1, HP 41CX handheld calculator (HP=Hewlett-Packard).
2, AC/DC power converters.
1, Wattmeter with electronic output and associated current transformers
2, HP 82181A extended memory module.
1, CMT-200 data acquisition and control unit.
(CMT=Corvallis MicroTechnology).
1, U of I/ARS A/D conversion (MAD) unit w/cables.
1, Weatherproof enclosure box, 18x12x8 inches. Any adequate size and type will do.
1, flow meter with electronic output.
1, pressure transducer (high) w/fittings.
1, pressure transducer (low) w/fittings.
1, plastic air line for low pressure bubbler.
1, flow meter/valve for low pressure bubbler.
Air supply with a regulator
Misc. pipe fittings for air line into water.
Misc. pvc conduits to protect buried air line.
1, CMT HPIL RAM Disk, model# RD-128, w/9-volt battery.
1, HP-IL MODULE, # HP-82160A.
1, HP THERMAL PRINTER, # HP-82162A, w/paper roll.
1, Portable carrying case for the RAM Disk, Thermal Printer, and IL Module. Should be waterproof and impact resistant, approx. 13x11x6 inches.
The monitor system manual

TOOLS AND SUPPLIES

Assorted Philips and flathead screwdrivers.
Electrical crimping/stripping tool.
Assortment of solderless connectors and wire nuts for wire gage sizes 18-22.
Soldering iron and solder.
Assorted wrenches and socket set, including a large pipe wrench (Minimum 3" jaw).
Assorted pliers and cutting dikes.

Cordless electric drill and bits. (optional)
2, 3, and 4 Conductor signal wire to be used between sensors and monitor box. Length depends on site.
Assorted conductor wires, sizes 18-22, and in colors red, black and green. Amounts depend on the number of sites and whether or not they are already wired.
Electrical tape and heat shrink.
1 to 2 foot long leads w/ alligator clips on each end.
Digital multimeter.
Spare 9 volt batteries.

HARDWARE INSTALLATION

Components

The MAD unit is attached to the enclosure box with two metal screws. Open the MAD unit and position it on the back plate of the enclosure box. Attach it with two screws and put the cover with the velcro strips back on.

Install the two EM modules in the two top ports of the calculator. The calculator should be turned off when you do this. Never remove the modules when the calculator is on. Also, be aware that even if the calculator is off, removing only one of the EM modules can cause all the data and information stored on both of them to be lost. For further information, consult the HP calculator and extended memory manuals.

Take the CMT-200 and calculator (w/EM modules) and install them in the grey enclosure box. Both units have velcro strips attached to their reverse sides which match the mating strips attached to the cover of the MAD unit. Connect the output module of the CMT-200 to one of the remaining available ports of the calculator. In this case it doesn't matter which of the remaining ports you choose.

Sensors

The pump efficiency monitor requires the installation of a minimum of four sensors for the primary pump performance parameters: input electrical power, flow rate, discharge pressure, and inlet water pressure or level. The sensor input leads in the enclosure box should be connected to the correct terminals of the MAD unit. You can

determine which one goes where by looking at the wiring diagram in figure 2. The exact wiring will depend on the types of sensors and MAD units used.

1. **Pulse emitting watt meters.**

   The pulse watt meter requires installation by either the power company or a certified electrician. A two conductor signal line is all that is necessary to connect the watt meter to the MAD unit. Consult with the electrician or power company to determine where to connect the lines to the watt meter. It will be necessary to have the electricity shut off to the pump station in order to install the current transformers (CTs). Have the electrician install the CTs on the main feed lines to the pump control panel. The location of the CTs and voltage taps determine which electrical components are monitored. Do not include building heaters, ventilation or auxiliary equipment, unless you want to include them in the pump station efficiency.

2. **Flow meter.**

   Two different types of signals from flow meters can be accommodated, pulse and analog. If existing flow meters do not have electronic output, most can be upgraded by contacting the manufacturor. The MAD unit accepts digital (pulse) signals or a 0-5 volt analog signal. See figure 2 for the correct wiring connections to the multiple analog to digital converter unit (MAD). The flow meter manual will assist in wiring the meter and meter installation.

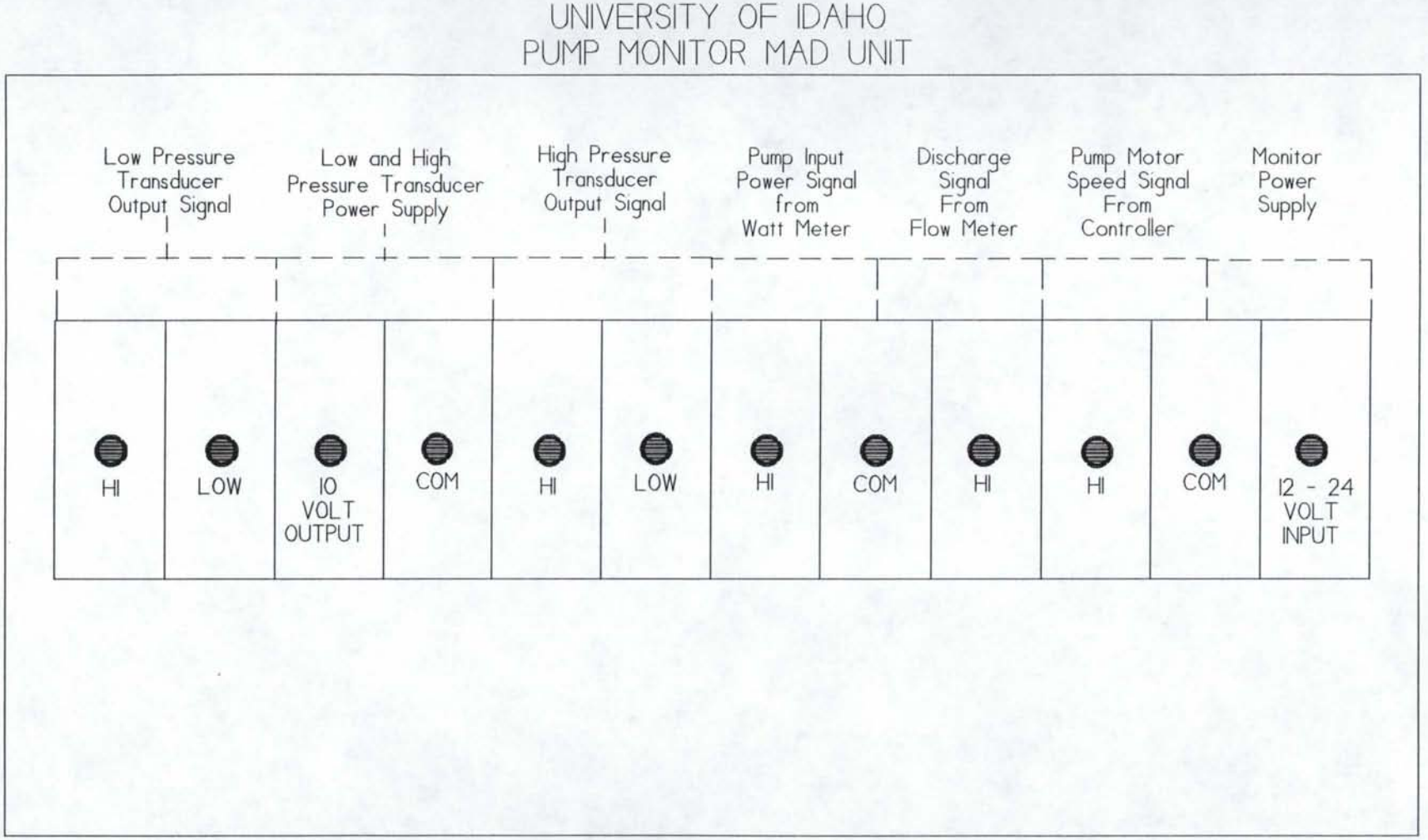

Figure 2. MAD Unit sensor connection diagram.

3. **High Pressure Transducer.**

The high pressure transducer is connected to the main pipeline. It is necessary to have the transducer isolated by two valves. The configuration of the valves should allow the transducer to be exposed to atmospheric pressure and to allow connection of a test gauge for calibrating the sensor. It may be necessary to drill a hole and weld on a 1/2" mounting plate if such a plate does not already exist on the present pipeline. Typically, the transducer will require a four conductor signal line to connect it to the MAD unit. The pressure transducer manual will aid in wiring and installation.

At this point, it may be well to survey the elevation difference between the tip of the air tube or inlet pressure sensor and point in the pipe where the high pressure transducer is mounted. Note this difference in your log book since it will be needed later when you initialize the monitor.

4. **Inlet Water Pressure or Level**

The inlet water pressure or level sensing system depends on the type of pump and its use. If the pump is functioning as a booster pump where the inlet pipe is under pressure, then a pressure transducer is all that is required. If the pump is a deep well pump, then the inlet water level sensing system requires an air tube, air supply, air control valves, air regulator, and a low pressure transducer. Depending on the transducer, a four conductor signal cable will be required to connect it to the MAD unit. The pressure transducer manual will assist in wiring and installation.

For booster systems, the pressure transducer is connected to the inlet pipeline. It is necessary to have the transducer isolated from the inlet pipe by two valves. The configuration of the valves should allow the transducer to be exposed to atmospheric pressure or to the water pressure inside the pipe and to allow connection of a test gauge for calibration of the sensor.

For well systems, the water level sensor system incorporates an air tube system to sense the depth to water from the high pressure sensor elevation. The actual measurement is the height of the water above the outlet (end) of the air tube in the well. The meter valve is used to control the rate of air release from the tank into the air tube. It should be placed in the line prior to the low pressure transducer. A needle valve is preferable. The transducer's pressure range depends on the maximum submerged length of the air tube.

PROGRAM CHECKLIST AND DESCRIPTION

The following is a list of the HP-41CX subroutines used in the pump station monitors.

EMAIN	ESCANF	SUMIT
AUTOES	SCANL	EINT
AUTOEL	SCANH	RAMSTO
ASCAN	EFCALC	ESTLOG
DSCAN	SITSTO	SPD
ESCANP		

These programs perform tasks as indicated below:

EMAIN: The control program. It has two entry points, AUTOS and AUTOL, which take care of automatic scanning and logging. It also controls the keyboard for user input and output. Many of the subroutines are called and executed by the EMAIN program.

AUTOES: Recalls site configuration constants, scans sensors, calculates flow, increments the number of scans, and stores data to intermediate registers. It calls several other subroutines to accomplish all this.

AUTOEL: Calculates summary data from the intermediate data and posts the summary data to extended memory.

ASCAN: Scans an analog channel, then converts the signal into its decimal equivalent. The value returned in the X-register.

DSCAN: Scans a digital channel for its signal, calculates the frequency or pulse rate. Frequency is returned in the X-register.

ESCANP: Scans watt meter. Uses DSCAN to obtain pulse rate, multiplies the frequency by the slope coefficient and adds the offset to obtain the power and stores it in register 4.

ESCANF: Scans flow meter. Uses DSCAN to obtain pulse rate or ASCAN to obtain the flow signal level, multiplies the resulting sensor value by the slope coefficient and adds the offset to obtain the flow, and stores it in register 5.

SCANL: Scans low pressure transducer. Uses the ASCAN routine to obtain the decimal reading from the low pressure transducer. Multiplies it by the slope coefficient and adds the offset to obtain the low pressure reading, and stores it in register 6.

SCANH: Scans the high pressure transducer. Uses the ASCAN routine to obtain the decimal reading from the high pressure transducer. Multiplies by the slope coefficient and adds the offset to obtain the high pressure reading, and stores it in register 7.

EFCALC: Recalls the current contents of the temporary storage registers containing the velocity head, discharge pressure head, input pressure head, elevation difference, flow rate, and input power. Uses these values and some constants to calculate the overall station efficiency.

SITSTO: Stores the site configuration constants and various pointers to extended memory.

RAMSTO: Transfers the data stored in extended memory to the portable RAM disk.

SPD: Scans pump speed. Uses ASCAN to obtain the speed signal, multiplies by the slope, and adds an offset to obtain speed.

SUMIT: Used to update intermediate data registers

EINT: The start-up, or initialization routine. It is used to read into the calculator all the necessary data coefficients, slopes, and offsets for each channel particular to a given site.

SETCLK: Used by the initialization routine to set the calculators internal clock to the correct time and date.

ESTLOG: Used by the initialization routine to set the calculator alarms for scanning and daily summary logging.

PROGRAM STORAGE AND LOADING

STORAGE (RAM DISK)

All the programs for the pump station monitor are stored on the RAM disk in one file. This disk has a storage capacity of 128K bytes. This is more than enough to store all the programs. In addition, station data files can be created that can store data transferred from the EM to the RAM disk. The RAM disk is powered by a 9 volt battery. Even if this battery loses power, an auxiliary battery ensures that the contents of the disk will not be lost.

At the beginning of the season the first thing you want to do with the RD (RD=RAM disk), is to make sure that it has a fresh 9 volt battery and auxiliary battery, and that the programs are still intact. To do this you will need:

The RAM disk.
The HP thermal printer (advisable but not necessary).
The HP IL module.
The HP 41CX calculator.

The connections are shown in figure 3; however, refer to the HP and CMT manual for in depth information. **CAUTION!!!** **ALL COMPONENTS MUST BE TURNED OFF WHEN MAKING OR BREAKING ANY CONNECTIONS.** Once the components are all properly connected, be sure that they are all turned on. If one component in the loop is not turned on, a NON-EXISTENT error message will appear on the calculator display. Now you are ready to check the RD directory. On the calculator keyboard press [XEQ] [ALPHA]. Then, using the blue lettered keys, type in {DIR}. Then press the [ALPHA] key again. If the printer is in the "norm" mode, you will get a printout of the directory at the same time that each program or data file is displayed on the calculator display. You can compare this printout with the previous one to make sure that all the programs are still there.

If you did not get a printout or directory listing on the calculator, do the following, otherwise go to the "Loading (calculator)" section below:

The loop connections are properly made.
All components are powered up.
The printer is in the norm mode.
You keyed the instructions in correctly.

If after doing this there is still no result, you will need to check the four hardware components. Try switching components with "fresh" ones one at a time. If there is still no result, perhaps the RD has gone completely dead and needs to be reformatted. To do this refer to Section 3 "Mass Storage Operations" on page 31 of the HP IL module owners manual. Any files that are on the RD will be lost.

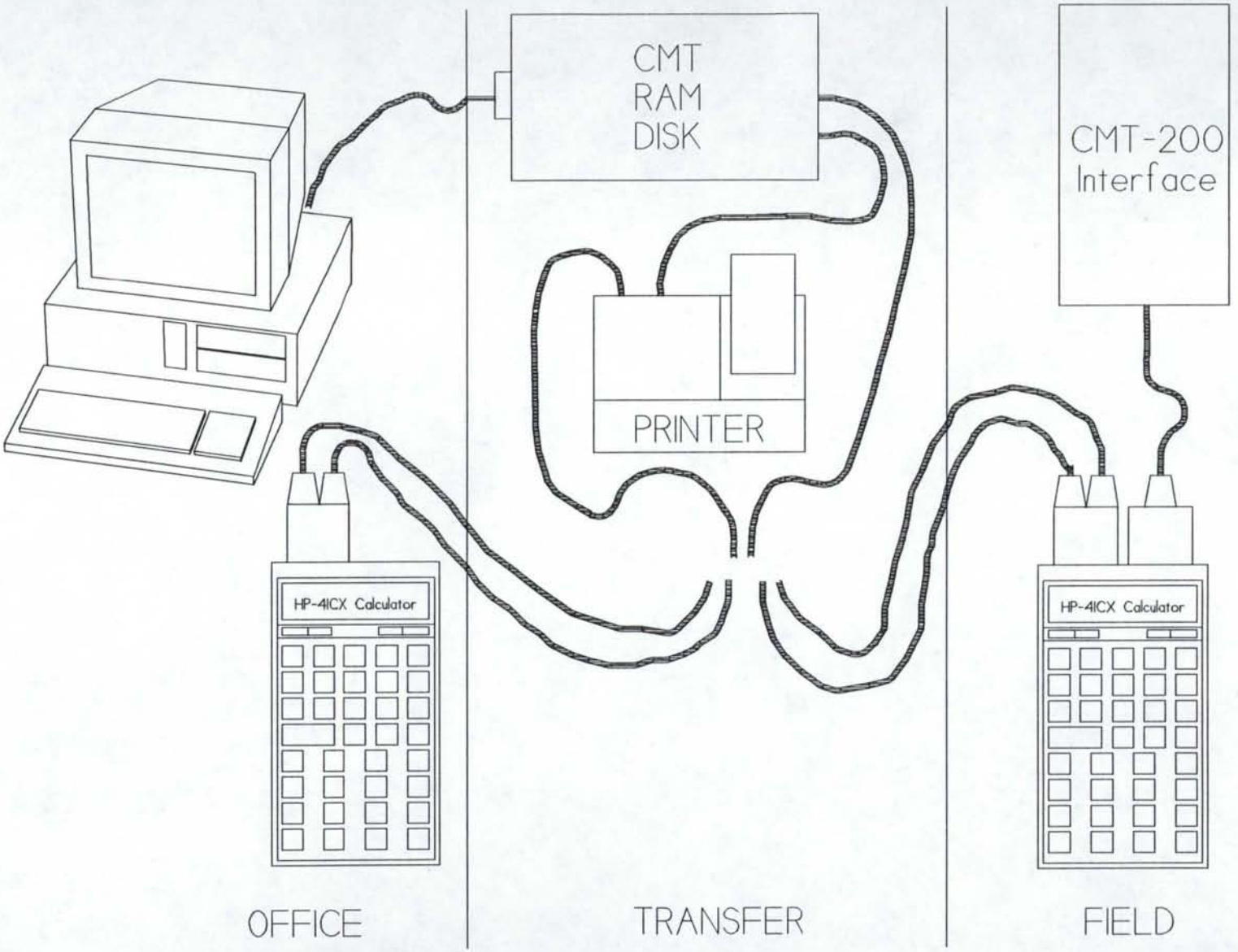

Figure 3. Pump station monitor data transfer equipment.

To load the RD with new programs you either need to find another RD or calculator with the programs already stored, or you will need to copy the programs line by line into the calculator main memory. In either case you need to understand the following:

The calculator is a controller.
The RD is simply a mass storage medium.
You need the calculator and HP IL module for input or output to the RD.

For input to the RD, a file or program must exist in the calculator main memory and have a specific file or program name. This name must be placed in the alpha register of the calculator. When you key in [XEQ] [ALPHA] {WRTP} [ALPHA] you can write the program specified in the alpha register to the RD.

For input from one RD to another, you again need the calculator as a controller. You first read a file or program from RD1 using the READA, READP, or READSUB commands into the calculator main memory. Turn off RD1 and the calculator and disconnect the calculator. Then you connect the calculator to RD2, power up both units, and use the WRTA or WRTP command as above to write this program into RD2. You can find detailed instructions on storing and retrieving programs on page 33 of the HP IL module owners manual.

The programs that are in the RD reside there permanently. As long as the main or auxiliary batteries are functioning, there should never be a need to copy each program line by line into the calculator main memory and then transfer them one by one into the RD. In case that ever happens, you can find complete program listings in Appendix 2.

This verification of programs on the RD should be carried out in the convenience of the office. Once you have verified the existence of all the programs that you will need on the RD, you should load them into the calculator in the field.

LOADING (CALCULATOR)

Be sure that the calculator has the EM modules installed. Turn it on to verify that you have power, then turn it back off. Press the back-arrow key and hold it down, then press the on key and release it, finally, release the back-arrow key. The message "MEMORY LOST" should appear in the calculator display. You have just cleared the calculator main memory and extended memory. The programs and data files will use all of the memory, that is why this step is necessary. Now turn the calculator back off. Connect the HP-IL unit to an available calculator port. The calculator and HP-IL should be in the loop with the RD and the printer. Turn all the units on. The calculator can be loaded program by program, or by loading a calculator "write-all" file. To load the calculator program by program perform the following steps:

1. [XEQ] [ALPHA] {SIZE} [ALPHA]. The calculator will prompt for a 3 digit input. Type in 060. This sizes the register allocation to 60 register blocks.

2. [GOLD KEY] [ASN] [ALPHA] {READSUB} [ALPHA]. Calculator will respond with a prompt for the key assignment. I usually use the blue "A" key in the top left of the keyboard. Instead of using [XEQ] [ALPHP] {READSUB} [ALPHA] every time you want to read a program from the RD to the calculator, all you have to do is put the name of the program into the alpha register, and press the blue "A" key. (You should have the calculator in "USER" mode.)

3. You should load the first program manually. Follow these directions.

   A. [ALPHA] {EMAIN} [ALPHA]. This loads the alpha register with the name of the program "EMAIN", which is the first program you want to put in the calculator.

   B. Now, [XEQ] [ALPHA] {READP} [ALPHA]. This keystroke series loads the program EMAIN from the RD to the calculator main memory. After you press the second [ALPHA] key, the display should go blank for a few seconds while the program is being read in. When the process is complete, whatever was in the X register last will be displayed, usually some number or name.

4. To load the next program (AUTOES in this case) do the following.

   A. [ALPHA] {AUTOES} [ALPHA], then press the blue "A" key, or the key assigned in step 2 (with the calculator in USER mode).

      The same thing should happen as before; a temporary blank display and then some number or name. Notice that this time you did not have to perform the keystroke series in step 3B above. That is why the key assignment was in step 2.

5. Continuing with AUTOEL program, load the remaining programs listed below.

PLACE IN THE ALPHA REGISTER	EXECUTE COMMAND
EMAIN	READP
AUTOES	READSUB
AUTOEL	READSUB
ASCAN	READSUB
DSCAN	READSUB
ESCANP	READSUB
ESCANF	READSUB
SCANL	READSUB
SCANH	READSUB
EFCALC	READSUB
SITSTO	READSUB
ESTLOG	READSUB
SPD	READSUB
SUMIT	READSUB
EINT	READSUB

6. Two things remain to be done at this point. Get rid of the key assignment (it uses up valuable main memory); and pack the main memory. To pack the main memory press [GOLD KEY] [GTO] [PERIOD] [PERIOD]. The display will flash the message "Packing".

To load the calculator with the "write-all" file, key in [ALPHA] {EMONTOR} [ALPHA] [XEQ] [ALPHA] {READA} [ALPHA]. The display will go blank while the file is being read into the calculator from the RD. The calculator is now loaded with all the necessary programs to operate the pump station monitor.

OPERATION

START–UP (INITIALIZATION)

Now that the calculator has all of the programs loaded, you need to tell it what kind of station it's dealing with, and many different station parameters. Also the internal clock and alarms need to be set. This is the initialization process. The program EINT, whose listing appears with the other programs listed in appendix B, performs the initialization. During the program execution the calculator display will prompt you with a series of questions.

Prior to starting the initialization process you will need to calculate all the slopes and offsets for all the sensors that you have installed in your pump monitoring system.

To help you do this, follow the steps below. First, let us define a few terms that were used in the equations.

Definitions:

R_a Register containing offset for the channel sensor

R_b Register containing slope for the channel sensor

f(x) MAD unit response for channel sensor

Y_1 First reading from the HP-41CX calculator

Y_2 Second reading from the HP-41CX calculator

ER_1 First external readings (i.e. from a pressure gauge)

ER_2 Second External reading

To determine the coefficients we will use the following mathematical expression.

$$Y= R_a+R_b*f(x)$$

$$R_a= ER_1-R_bY_1$$

$$R_b= (ER_2 - ER_1) /(Y_2 - Y_1)$$

To begin make sure that your pump and HP-41CX monitoring system are fully operational. Start the pump and also turn on the monitor. You will need to execute EMAIN in the monitor, and then you will need to initialize the system. When you come to the part in the initialization where it requests slopes and offset you will enter 0 in for the offset and 1 in for the slope. This will allow the calculator to operate and will return MAD unit responses, f(x), for the sensor channels.

For example, we will determine the coefficients for the discharge pressure. The pump is on and you have completed the initialization of the monitoring system. At this time press the "D" key on the calculator and wait for it to return a number to the display. When the number appears record that number, at the same time observe the reading on the external pressure gauge and record these numbers as a pair. Repeat this procedure again. This time you will need to reconfigure the pump's operation to obtain

different values than the first time. For this example we will assume that you just completed the procedure and obtained the following values.

ER_1 = 240 psi - Pump operating

ER_2 = 100 psi - Pump not operating

Y_1 = 0.75

Y_2 = 0.02

Plugging these numbers into the equations will yield the following coefficients. Calculate Rb first so that you can use that value to determine R_a.

R_a = REGISTER 18 - High pressure offset

R_b = REGISTER 19 - High pressure slope

R19 = $(ER_2 - ER_1) / (Y_2 - Y_1)$
= (240 - 100)/ (0.75 - 0.02)
= 191.78

R18 = $ER_1 - R19*f(x)_1$ (R19 = 191.78)
= 240 - (191.78 * 0.02)
= 236.16

R18 = offset
= 236.16

R19 = slope
= 191.78

Now that you have determined the coefficients, enter them in the appropriate spaces on the blank initialization form in Appendix 1. The output/input elevation difference is the measured distance between the high pressure transducer and the point where the water level is being calculated from. You should have measured this when you were installing the pressure transducers.

After you have entered the information for the last prompt, the display should show "DONE" after several seconds. You now have finished the initialization process. The calculator will "wake-up" at the time and date you specified and begin scanning sensors, make calculations and storing data.

OPERATIONAL CHECKS

There are several things that you can check to make sure that the program is operating correctly once it is up and running.

Power–up

Make sure you have power. When you turn the calculator on, there should be a beep, followed by the current time and date. If the current time and date are incorrect, you should first dump any data (see the "Data Transfer to the RAM Disk" section on page 21), and then re-initialize (see the "Start-up (Initialization)" section on page 15. In any case, in the lower right corner of the display you should also see the program annunciator "PRGM". This tells you that the calculator is in program mode. It must be in program mode in order to execute any of it's routines. If it is in program mode and the EMAIN program is working correctly, it should turn itself off ninety seconds after the last operation it performed.

Information Display

This feature allows you to check some current values of selected station parameters. It is a built in routine of the main program, and thus does not require that you break in. When the calculator is in program mode you can press the blue "A, B, C, D, E, F, G, H, or I" keys to accomplish the following:

A= Display of current input horsepower.
B= Display of current flowrate (cfs).
C= Display of current input pressure (ft of head).
D= Display of current output pressure (ft of head).
E= Display of current efficiency (%).
F= Pump speed (rpm).
G= Display of current pumping costs ($/AF).
H= Executes initialization routine.
I= Dumps data from EM to RAM disk.

It may take anywhere from 5 seconds to a couple of minutes for the calculator to display the information you request.

Read and Write Pointers

These pointers simply are an accounting tool of the calculator to keep track of how many of the registers in EM have been filled and/or emptied, that is, written to or read from respectively. EM is 560 registers long and works like a circular file. That is, when register 560 is finally filled, the next register to be filled will be number 001. Whatever was in register 001 will be written over. The write pointer tells you up to what register information has been stored. The read pointer tells you up to what register information has been read and stored on the RAM disk, RD.

The pump monitor program gives you the flexibility to set the number of times you scan and log data. You must be careful and make sure you have calculated the number of days that the system can store your data before you start writing over your records. For example, we will assume that you have 560 registers available to store data in. This program stores data to the extended memory in register blocks of 7 registers. If you were to set your scan interval at 15 minutes and log every hour, you would then have used 7 registers every hour. At the end of one day you would have used 168 registers. This would leave you approximately two days before you would have to dump the data to the RAM disk or data would start being overwritten.

BREAKING IN

During normal operation of the monitor the program EMAIN is in control of all program executions together with the "wake-up" alarms. Regardless of which subroutine might be operating at any given time, control always returns to the EMAIN program. At all such times, the "PRGM" annunciator will appear in the lower right corner of the display. If you press any key except the nine keys which were discussed earlier, you will invoke an error message telling you "INVALID KEY". This prevents you from inadvertently altering any of the programs or stored data. However, you can get out of program mode and access the programs and EM by BREAKING IN. This you do by

pressing the ENTER key. When you do you will notice that the PRGM annunciator vanishes. Now you can operate the calculator from the keyboard. This gives you the flexibility to recall registers and make program and coefficient changes whenever necessary.

Getting back to R&W pointers - once you have broken in, simply press [RCL] {12}. This recalls memory register 12, the register in which the write pointer is stored. Refer to register map section (Appendix 2) for a complete pump monitor program register map. If the value in the register is satisfactory, all you need to do to get back into program mode is press the R/S key. If you want to change the number say to 56, press {56} [STO] {12}. This stores a new value of 56 in register 12. Next, press [XEQ] [ALPHA] {SITSTO} [ALPHA]. This stores your register change to EM, from where it is recalled during subsequent scannings. After this operation, you need to press the following keys: [XEQ] [ALPHA] {EMAIN} [ALPHA]. Recapping the steps you've taken:

1. Break-in. [ENTER]
2. Check some register. [RCL] {xx}
3. Make some change. i.e {56} [STO] {12}
4. Save change to EM. [XEQ] [ALPHA] {SITSTO} [ALPHA]
5. Get back into monitor mode. [XEQ] [ALPHA] {EMAIN} [ALPHA]

So now you know how to BREAK-IN so that you can check and alter, if necessary, your R&W pointers. You can also use the break-in feature to check the time, date and alarms.

DEPARTURE

The last operational check that you should make before leaving any site is to make sure that the calculator is in program mode. If you happen to be there when it is due to make its next scan, you could also watch to see that it wakes up and scans. (a pointer will move back and forth across the display) The calculator will then turn itself off after 90 seconds. Remember, make sure that the calculator is in monitor mode before you leave the site.

DATA TRANSFER TO THE RAM DISK

The electronic transfer of data from the calculator EM to the RD is straight forward. Follow these steps:

A. Connect the calculator, HP-IL module, RD and the printer in the series loop, figure 3.
B. Turn all the components on.
C. Press the "I" key.

After a few seconds the transfer should commence. The printer will print the results while the transfer is being made. It will first print the station name and today's date. Then the data follows in the following format:

R50= MDD.HHMMSS	DATE AND TIME
R51= ##.#######	NUMBER OF SUCCESSFUL OBSERVATIONS
R52= ###.#######	INPUT HORSEPOWER (HP)
R53= #.#######	CURRENT FLOW (CFS)
R54= ###.#######	WATER LEVEL (FT)
R55= ###.#######	DISCHARGE HEAD (FT)
R56= ####.#######	PUMP SPEED (RPM)
R57=##.#######	PUMP EFFICIENCY (%)

The description of the data is in the right hand column. This information does not appear on the printout. When the transfer is completed, the word DONE will appear in the calculator display and will be printed by the printer. Here are some trouble shooting tips:

If after pressing the "I" key nothing happens, or you get an error message such as "NONEXISTENT", check the following:

a. Connections and power-on for all components.
b. Make sure that the calculator is in program mode.
c. R&W pointers in the calculator should be different values. Write pointer should have advanced. If not, you will get a "DONE" message printed by the printer and in the calculator display, but no data.
d. Run Cat 1 for the calculator. This displays all programs in the calculator main memory. Be sure that RAMSTO is listed. If not, load it in! The transfer cannot take place without that program.

If the transfer stops before the word DONE appears, you will probably get an error message such as NONEXISTENT, or END OF FILE, or RAM FULL. You can check several things to remedy this.

a. Printer out of paper.
b. Printer not sufficiently charged.
c. RAM Disk battery is low.
d. Station file on the RD is not of sufficient length to accommodate the number of days you are trying to dump. In this case the message RAM FULL will be displayed. The PCMTSTO program will update the calculator read pointer and the RAM write pointer up to (but not including) the last day printed. These days are also stored on the RAM DISK. Any additional days of data will not be dumped. You will need to reformat the RAM. Refer to the Office Manual to accomplish this.

Now you will have a hard copy of the data transfer, the data will be stored on the RAM, and the RAM write pointer will have advanced 7 times the number of days you dumped. If you check the calculator R&W pointers after the transfer the read pointer should be the same value as the write pointer.

DATA TRANSFER FROM RAM TO PC

To transfer data collected by the pump monitor and upload to the RAM disk unit to a PC, the following items will be needed:

HP-41CX calculator and HP-IL (interface loop)
RAM disk
IBM compatible PC with serial port and GW-BASIC
Serial cable and connectors
HP to PC basic programs (see Appendix 3)

After the basic programs have been loaded into the PC, you should make the following connections (see figure 3). Make sure that calculator, RAM disk, PC, and printer are turned off before connecting.

1. The RAM disk should be connected to the HP-41CX calculator by the HP-IL cable.

2. Connect the serial port of the PC to the RAM disk using the serial cable.

3. Verify connections and turn on the PC and HP-41CX.

4. Execute the BASIC program, "HPINPUT0" on the PC. You should see a title screen and instructions - - follow them.

5. To start the HP-41CX data dump, enter the following keystrokes: [XEQ] [ALPHA] {RAMDUMP} [ALPHA].

At the end of the transfer, the RD pointers will be updated.

APPENDIX 1 – PUMP INITIALIZATION WORKSHEET

To the user: If you will take the time to fill out this worksheet you will find that it is in the same format as the calculator will prompt you for input during the initialization process.

STATION NAME: _____
Enter the station name using up to 6 characters

NOTE: Turn to the appendices for instructions on how to calculate IP, OP coefficients

I-A?: _____
The calculator is asking you for the water level low pressure offset, ft

I-B?: _____
The calculator is prompting you for the water level low pressure slope, ft

D-A?: _____
The calculator is prompting you for the discharge pressure offset, ft head

D-B?: _____
The calculator is prompting you for the discharge pressure slope, ft head

O-I ELEV? _____
The calculator is prompting you for the output / input elevation difference.

S-A: _____
The calculator is prompting you for the speed sensor offset, rpm

S-B: _____
The calculator is prompting you for the speed sensor slope, rpm

MAX EFF? _____
The calculator is prompting you for the pumps rated maximum efficiency. Enter the number in the decimal form of percent (i.e. 65% = 0.65)

WM-A: _____
The calculator is prompting you for the watt meter offset

WM-B: _____
The calculator is prompting you for the watt meter slope.

KWH COST? _____
The calculator is prompting you for the current energy cost which can be obtained from the local utility company, $/kw

PA: _____
The calculator is prompting you for the pipe area, sq ft.

SCAN H.MS? _____
The calculator is prompting you for the scan interval. EXAMPLE: For a scan interval of every 30 minutes you would enter 0.3000

LOG H.MS _____
The calculator is prompting you for interval to summarize and log the scanned data to EM. For example, logging interval of 4 hours would be entered as 4.000.

START M.DY? _____
The calculator is prompting you for the *Month,Day,Year* that you want the pump monitor to start data collection. EXAMPLE: If you wanted the pump monitor to start on January 1, 1999 you would enter 1.11999.

START H.MS? _____
The calculator is prompting you for the Hour,minute,second that you want the scan interval to begin. EXAMPLE: If you wanted to start at 1:30 pm, you would enter 13.0000 (Note: The time format is in 2400 hr.)

APPENDIX 2. HP41 pump monitor program listings, register maps and flag useage.

HP41 PUMP STATION MONITOR REGISTER MAP

Register Number	Contents of the register	Units
1		
2		
3	Registers 00 thru 09 are used by several	
4	of the subprograms in EMAIN. Depending on	
5	which subprogram is being used at the time	
6	will determine the contents of the	
7	register(s).	
8		
9		
10	Station name (i.e., Ketchum or Sun Valley)	
11	Extended Memory Read Pointer	registers
12	Extended Memory Write Pointer	registers
13	Speed sensor offset	rpm
14	Speed sensor coefficient	rpm/f(x)
15	Output - Input elevation difference	ft
16	Water level offset	ft
17	Water level coefficient	ft/f(x)
18	Discharge Pressure offset	ft
19	Discharge Pressure coefficient	ft/f(x)
20	Number of Flow meters	
21	Flow meter offset	cfs
22	Flow meter coefficient	cfs/f(x)
23	Pipe area	sq-ft
24		
25	Extended Memory file size	registers
26		
27		
28		
29		
30	Number of watt meters	
31	Watt meter offset	hp
32	Watt meter slope	hp/f(x)
33		
34		
35		
36		
37		
38		
39	Energy cost	$/kwhr
40	Date and time	
41	Number of observations during a scan interval	
42	Total input horsepower during a scan interval	hp
43	Total flow during a scan interval	cfs
44	Total water level during a scan interval	ft
45	Total discharge pressure during a scan interval	af
46	Total efficiency during a scan interval	%
47		
48		
49		
50	Transfer date & time to ram disk	
51	Transfer number of observations to ram disk	
52	Transfer calculated input horsepower to ram disk	hp
53	Transfer calculated flow to ram disk	cfs
54	Transfer water level to ram disk	ft

55	Transfer calculated discharge pressure to ram disk	ft
56	Transfer calculated speed to ram disk	rpm
57	Transfer calculated efficiency to ram disk	%
58		
59		

HP41 PUMP STATION MONITOR CALCULATOR FLAG USAGE

A flag has only two states, set or clear. These states can be interpreted as "on/off" (like a switch), as "yes/no" (like a decision). The calculator has 56 flags. Also the first eight flags (00 thru 07) can be interpreted as the eight bits in a byte, and that byte can be transformed into a number in the x register.

FLAG MAP

Flag Number	Condition
1	Pump on, using power and flowing?
2	Water level scanned?
3	Efficiency calculated?
4	Hp scanned?
5	Flow scanned?
6	Water level (yes)
7	Discharge pressure (yes) Is there discharge pressure at this time?
8	
9	
10	Set if pump is running

HP41 PUMP STATION MONITOR FIELD PROGRAM LISTINGS

Main Control Program

Program Name: EMAIN

```
01        LBL "EMAIN"          Global label for main control program
02        GTO 00
03        LBL "AUTOS"          AUTOMATIC SCANNING ENTERY POINT
04        XEQ "AUTOES"         AUTOMATIC SCAN OF PUMP SENSORS
05        GTO 99
06        LBL "AUTOL"          AUTOMATIC DAILY DATA POSTING
07        XEQ "AUTOEL"         AUTOMATIC POSTING OF DATA REGISTERS
08        GTO 99

                               THE USER WANTS THE INPUT HORSEPOWER NOW
                               PRESS THE KEY WITH "A" ON IT
09        LBL 11
10        XEQ "ESCANP"         SUBPROGRAM THAT CALULATES INPUT HORSEPOWER
11    GTO 00

                               THE USER WANTS THE FLOW NOW.
                               PRESS THE "B" KEY or 1/X
12        LBL 12
13        XEQ "ESCANF"         SUBPROGRAM THAT CALULATES FLOW (CFS)
```

```
14      GTO 00

                        THE USER WANTS INPUT/HEAD OR WATER LEVEL
                        PRESS THE "C" KEY or SQUARE ROOT

15      LBL 13
16      XEQ "LNOW"
17      GTO 00

                        THE USER WANTS THE DISCHARGE PRESSURE
                        PRESS THE "D" KEY or LOG
18      LBL 14
19      XEQ "HNOW
20      GTO 00
                        THE USER WANTS THE PUMPING STATION EFFICIENCY
                        NOW
                        PRESS THE "E" KEY or LN
21      LBL 15
22      XEQ "ENOW"      SUBROUTINE THAT CALCULATES EFFICIENCY
23      GTO 00

                        THE USER WANTS THE PUMP SPEED NOW
                        PRESS THE "F" KEY or X<>Y
24      LBL 21
25      XEQ "SPD"
26      GTO 00

                        THE USER WANTS THE CURRENT PUMPING COST
                        PRESS THE "G" KEY
27      LBL 22
28      XEQ "CNOW"
29      GTO 00

                        THE USER WANTS TO INITILIZE THE SYSTEM
                        PRESS THE "H" KEY or SIN
30      LBL 23
31      XEQ "EINT"
32      "DONE"
33      GTO 00

                        THE USER WANTS TO DUMP THE DATA FROM THE
                        CALCULATOR TO THE RAM DISK

34      LBL 24
35      XEQ "RAMSTO"    GO DUMP THE DATA TO THE RAM DISK
36      LBL 00
37      AVIEW
38      FIX 2           FIX 2 DECIMAL PLACES
39      XEQ BEEP        WILL CAUSE THE CALCULATOR TO BEEP
40      99              WAIT 99 SECONDS
41      GET KEY X       GET USERS REQUEST
42      CLA             CLEAR DISPLAY
43      X<>Y            EXCHANGE  X REGISTER WITH Y REGISTER
44      4
45      X<>Y
46      X<=Y
47      GTO 99
48      41              BREAK PROGRAM KEYCODE
49      X=Y?            TEST FOR THE ENTER KEY
50      GTO 98          TURN ON THE NORMAL KEYBOARD
51      X<>Y            BRING BACK THE KEYCODE FOR ANOTHER TEST
```

```
52        25                    MAXIMUM KEYCODE ACCEPTED
53        X>Y?                  CHECK FOR A VALID USER RESPONSE
54        GTO IND Y             GO DO WHAT IS REQUESTED
55        "INVALID KEY"         ERROR MESSAGE TO USER
56        GTO 00                RETURN BACK TO ENTERY POINT
57        LBL 99                ADDRESS OF SUBROUTINE TO POWER DOWN THE HP 41
                                CALCULATOR
58        0                     POWER DOWN THE SYSTEM
59        X<>F
60        SF 11                 SET CONTINUE FLAG
61        OFF                   SHUT THE CALCULATOR OFF
62        XEQ "SITRCL"          RECALL SITE CONFIGURATION DATA
63        CLA                   CLEAR CALCULATOR DISPLAY
64        DATE                  SHOW TODAYS DATE
65        ADATE                 APPENDS CURRENT DATE IN ALPHA REGISTER
66        APPEND
67        TIME                  RECALL THE CURRENT TIME FROM THE CALCULATOR
68        ATIME                 SHOW ME THE CURRENT TIME
69        GTO 00
70        LBL 98                SHUT DOWN THE LOGGING SYSTEM
71        STOP
72        GTO 00
73        END
```

PROGRAM NAME: AUTOEL - - USED TO POST SUMMARY REGISTERS

```
01        LBL "AUTOEL"          GLOBAL ENTERY POINT LEVEL
02        XEQ "SITRCL"          RECALL SITE CONFIGURATION REGISTERS
03        DATE                  SHOW A NUMERIC EQUIVALENT OFTODAYS DATE
04        100
05        *
06        INT                   RETURN AN INTEGER PART OF A NUMBER
07        TIME                  RETURN NUMBER FOR CURRENT TIME
08        100
09        /
10        +
11        STO 40                STORE IT IN REGISTER 40
12        RCL 41                RECALL THE NUMBER OF SUCCESSFUL SCANS
13        X=0?                  DOES THE NUMBER IN REG. 41 EQUAL ZERO? SKIP THE
                                NEXT LINE UNLESS X EQUALS ZERO.
14        GTO 01                GOTO LABEL 01
15        042.047
16        STO 48                STORE THE NUMBER FOR THE REGISTER BLOCKS 042.047
17        LBL 00
18        RCL 41                RECALL THE NUMBER OF SUCCESSFUL SCANS
19        ST IND 48
20        ISG 48
21        GTO 00
22        LBL 01
23        ARCL 10               RECALL THE STATION NAME
24        35                    DECIMAL CODE FOR A NUMBER
25        XTOA                  CONVERT THE NUMBER IN THE X REGISTER TO
                                EQUIVALENT BYTE AND PUT IT IN THE ALPHA REGISTER
26        RCL 12                RECALL THE EXTENDED MEMORY WRITE POINTER
27        FILESIZE              RETURN THE NUMBER OF FILES TO THE DISPLAY
28        MOD                   ADJUST THE POINTER
29        SEEKPTA               SET THE FILE POINTER
30        040.047               REGISTER BLOCK
31        SAVERX                SAVE THE REGISTER BLOCK TO THE FILE
32        8                     LOAD X WITH THE RECORD LENGTH IN THE REGISTER
```

```
33      ST +12              TAKE THE NUMBER IN THE X REGISTER AND ADD IT
                            THEN STORE IT IN REGISTER 12
34      040.047             REGISTER BLOCK IN MAIN MEMORY
35      CLRGX               CLEAR THE CONTENTS OF THE REGISTER BLOCK
36      XEQ "SITSTO"        UPDATE THE SITE CONFIGURATION FILE
37      RTN
38      END
```

PROGRAM NAME: AUTOES - CONTROL THE AUTOMATIC SCANNING OF THE PUMP SENSORS

```
01      LBL "AUTOES"        GLOBAL ENTERY LABEL
02      XEQ "SITRCL"        RECALL SITE CONFIGURATION DATA
03      LBL 00
04      0                   CLEAR
05      X<>F                CLEAR ALL FLAGS SET
06      XEQ "ESCANP"        GO GET THE CURRENT FLOW
07      XEQ "ESCANF"        GO GET THE INPUT HORSEPOWER
08      FC? 00              IS FLAG 00 CLEAR
09      GTO 01
10      XEQ "LNOW"          GO GET THE WATER LEVEL
11      XEQ "HNOW"          GO GET THE DISCHARGE PRESSURE HEAD
12      XEQ "SPD"           GO GET THE PUMP SPEED
13      XEQ "EFCALC"        GO GET THE PUMP'S CURRENT EFFICIENCY
14      RCL 27              RECALL THE MAXIMUM EFFICIENCY
15      RCL 09              RECALL THE CURRENT EFFICIENCY
16      X<=Y?               IF THE NUMBER IN REGISTER 27, Y, IS GREATER THAN
                            THE NUMBER IN REGISTER 09 SKIP THE NEXT
                            INSTRUCTION
17      GTO 01              GO TO LABEL 01
18      ISG 26              CHECK THE NUMBER IN REGISTER 26. FOR A NUMBER
                            (ii.jjkk) IN REGISTER 26,INCREMENTS ii by kk AND
                            SKIPS THE XT PROGRAM LINE IF ii+kk>jj.
19      GTO 00
20      LBL 01
21      RCL 27              RECALL THE MAXIMUM EFFICIENCY
22      STO 26              STORE THAT NUMBER IN REGISTER 26
23      FC? 10
24      XEQ "SUMIT"         GO ADD THE DATA TO THE APPROPRIATE REGISTERS
25      RTN
26      END
```

```
PROGRAM NAME:  ASCAN - - SCANS THE ANALOG CHANNELS

01        LBL "ASCAN"          GLOBAL ENTERY LEVEL
02        INVON                INVERT MODE ON
03        ODX                  THIS FUNCTIONS SETS THE 8 OUTPUT LINES ACCORDING
                               TO THE BINARY EQUIVALENT OF THE INTEGER IN THE X
                               REGISTER. FOR EXAMPLE IF THE NUMBER 130 IS IN
                               THE X REGISTER THE OUTPUT WOULD BE AS FOLLOWS:

                               1=CLOSED CIRCUIT   0=OPEN CIRCUIT
                               LINE #
                                        STATE
                               D07          1
                               D06          0
                               D05          0
                               D04          0
                               D03          0
                               D02          0
                               D01          1
                               D00          0
04        2                    INPUT PRESSURE CONVERSION
05        -
06        ODX                  OUTPUT DATA FROM X
07        2                    INPUT PRESSURE START CONVERSION
08        +
09        ODX                  OUTPUT DATA FROM X
10        LBL 00
11        ICX                  READS THE INPUT CONTROL LINE AND PLACES THE
                               RESULT (1 OR 0) IN THE X REGISTER
12        X NE 0?              CHECK THE DONE STATUS
13        GTO 00
14        18                   GET THE DATA CODE FOR THE OUTPUT
15        ODX                  TELLS THE MADD UNIT TO SEND IT
16        IDX                  GET A VALUE
17        256
18        /                    TRANSFORM THE NUMBER TO A DECIMAL NUMBER BY
                               DIVIDING BY 256
19        RTN                  RTN BACK WITH THE READING IN THE X REGISTER
20        END


PROGRAM NAME:  DSCAN

FUNCTION:  SCAN DIGITAL LINES

01        LBL "DSCAN"          GLOBAL ENTERY LEVEL
02        INVON                THIS FUNCTION INVERTS THE BITS OF EVERY BYTE
                               SENT TO THE OUTPUT LINES D00-D07.
03        3                    DIGITAL DATA INSTRUCTION CODE FOR THE MADD UNIT
04        ODX                  OUTPUT DATA FROM X
05        2                    BUILD INPUT LINE MASK
06        RCL Z                RECALL THE DIGITAL LINE TO SCAN
07        y^X                  RAISE Y TO THE X POWER
08        1500                 BUMP MASK TO Y AND LOAD THE X REGISTER WITH 15
                               SECONDS
09        RATE                 COUNT THE NUMBER OF EVENTS IN THE TIME
                               INTERVAL X
10        X=0?                 IF NO PULSES WHERE COUNTED
11        RTN                  RETURN
12        30
```

```
13        X<=Y?
14        GTO 05
15        RDN                    MOVE STACK DOWN
16        RCL Z                  RECALL THE DIGITAL LINE TO SCAN
17        20                     NEED 20 BYTE TIME BUFFER
18        BUFX                   CREATE A BUFFER OF 20 BYTES
19        X<>Y                   BRING BACK THE MASK
20        8                      WILL RECORD THE TIME FOR 8 STATE CHANGES
21        TIMEI                  TIME THE INPUT SIGNAL
22        2                      WILL USE THE TIME INTERVALS BETWEEN THE 2nd AND
                                 8th
23        X>PT                   USE X TO SET THE INPUT/OUTPUT BUFFER
24        1.006001
25        STO 03
26        0                      PUT A ZERO IN THE X REGISTER FOR SUMMATION
27        LBL 00
28        BUF>TX                 COPY THE NEXT TWO BYTES IN THE BUFFER AS TIME
                                 DATA TO X
29        +
30        ISG 03
31        GTO 00                 GO DO IT AGAIN
32        300                    2 STATE CHANGES /PULSE AND 1 SEC/100
                                 (CENTISECOND)
33        /
34        1/X
35        RTN
36        LBL 05
37        /
38        RTN
39        END
```

PROGRAM NAME: ESCANP

FUNCTION: PERFORM THE POWER SCAN ON THE WATT METER

```
01        LBL "SCANP"            GLOBAL ENTER LEVEL
02        "NO-HP"                TELL THE USER THAT THE PUMP IS OFF
03        0                      PUT ZERO IN THE X REGISTER
04        STO 04                 PUT ZERO IN REGISTER 04
05        XEQ "DSCAN"GO _______SCAN THE DIGITAL LINES
06        X=0?                   DOES THE NUMBER EQUAL ZERO?
07        RTN
08        RCL 32                 RECALL THE WATT METER SLOPE
09        *
10        RCL 31                 RECALL THE WATT METER OFFSET
11        +
12        STO 04                 STORE IT IN REGISTER 04
13        SF 04                  SET THE FLAG TO INDICATE THAT THE PUMP IS ON
14        SF 00
15        "HP-IN:"
16        ARCL 04                RECALLS THE INPUT HORSEPOWER AND PUT IT IN THE
                                 ALPHA DISPLAY  (ie. HP-IN:98.654889)
17        RTN
18        END
```

PROGRAM NAME: ESCANF

FUNCTION: PEFORM FLOWRATE SCAN ON THE HIGH AND LOW PRESSURE TRANSDUCERS

```
01      LBL "ESCANF"        GLOBAL ENTERY LEVEL
02      "NO-Q"
03      0                   PUT A ZERO IN THE X REGISTER
04      STO 05              CLEAR REGISTER 05
05      1
06      XEQ DSCAN           GO SCAN THE DIGITAL LINE
07      X=0?
08      RTN
09      RCL 22              RECALL THE FLOW METER SLOPE
10      *
11      RCL 21              RECALL THE FLOW METER OFFSET
12      +
13      STO 05              STORE IT IN REGISTER 05
14      SF 03
15      SF 00
16      "Q-CFS:"
17      ARCL 05             RECALL THE FLOW AND PUT IT IN THE DISPLAY (ie
                            Q-CFS: 1.788986)
18      RTN
19      END
```

PROGRAM NAME: SCANL

FUNCTION: CALCULATE THE INPUT PRESSURE HEAD

```
01      LBL "SCANL"         GLOBAL ENTER LEVEL
02      23                  REGISTER WITH FLOW METER OFFSET
03      XEQ "ASCAN"         GO SCAN THE ANALOG CHANNELS
04      RCL 17              RECALL WATER LEVEL SLOPE
05      *                   MULTIPLY FLOW METER OFFSET BY THE WATER LEVEL
                            SLOPE
06      RCL 16              RECALL WATER LEVEL OFFSET
07      +                   ADD THE WATER LEVEL OFFSET
08      STO 06              STORE IT IN REGISTER 06
09      SF 02               SET FLAG 02 TO INDICATE A SCAN ON THE WATER
                            LEVEL
10      RTN                 THATS ALL
11      END
```

PROGRAM NAME: SCANH

FUNCTION: CALCULATE THE DICHARGE PRESSURE HEAD

```
01      LBL "SCANH"         GLOBAL ENTERY LEVEL
02      27                  REGISTER WITH MAXIMUM PUMP EFFICIENCY
03      XEQ "ASCAN"         GO GET THE INPUT SENSOR READING
04      RCL 19              RECALL DISCHARGE PRESSURE SLOPE
05      *                   MULTIPLY THE MAXIMUM EFFICIENCY BY THE DISCHARGE
                            PRESSURE SLOPE
06      RCL 18              RECALL DISCHARGE PRESSURE OFFSET
07      +                   ADD THE DISCHARGE PRESSURE OFFSET
08      STO 07              STORE THAT NUMBER IN REGISTER 07
09      SF 01               SET FLAG 01. TELLS THE CALCULATOR THAT THE PUMP
                            IS RUNNING
10      RTN                 YOUR DONE
```

```
11        END
```

PROGRAM NAME: EFCALC

FUNCTION: CALCUALTE THE PUMP EFFICIENCY

```
01        LBL "EFCALC"          GLOBAL ENTERY LEVEL
02        RCL 05                RECALL VELOCITY HEAD
03        RCL 23
04        /
05        X^2                   RAISE X TO THE SECOND POWER
06        64.4                  2 TIMES GRAVITY  (2G)
07        /
08        RCL 07                RECALL OUTPUT PRESSURE HEAD
09        +
10        RCL 06                RECALL INPUT PRESSURE HEAD
11        -
12        RCL 15                RECALL THE OUTPUT/INPUT SENSOR ELEVATION
                                DIFFERENCE
13        +
14        RCL 05                RECALL THE FLOW
15        *
16        RCL 04                RECALL THE INPUT HORSEPOWER
17        /
18        11.34545              62.4 DIVIDED BY 550 TIMES 100
19        *
20        STO 09
21        SF 09
22        RTN
23        END
```

PROGRAM NAME: SITSTO

FUNCTION: TRANSFERS SITE DATA FROM THE MAIN MEMORY TO THE EXTENDED MEMORY

```
01        LBL "SITSTO"          GLOBAL ENTERY LEVEL
02        "LOGGER"              SITE DATA FILE NAME
03        0                     PUT A ZERO IN THE X REGISTER TO SET THE FILE
                                POINTER TO THE BEGGING OF THE FILE
04        SEEKPTA               SET THE FILE POINTER TO ZERO
05        10.039                REGISTER BLOCK TO BE COPIED TO THE FILE
06        SAVERX                SAVE THE REGISTERS
07        LBL "SITRCL"
08        "LOGGER"              SITE DATA FILE NAME
09        0
10        SEEKPTA               SET THE FILE POINTER TO ZERO
11        10.039
12        GETRX                 COPY THE REGISTER BLOCK
13        RTN
14        END
```

PROGRAM NAME: ESETLOG

FUNCTION: TO SET THE ALARMS FOR THE WANTED SCAN INTERVALS

```
01      LBL "ESETLOG"          GLOBAL ENTERY LEVEL
02      CLRALMS                CLEAR ALL PREVIOUS ALARMS
03      "SCAN H.MS?
04      PROMPT                 TELLS THE CALCULATOR HOW MANY TIMES YOU WANT THE
                               MONITOR TO PERFORM SCANS
05      STO 01                 STORE THAT NUMBER IN REGISTER 01
06      "START M.DY?"
07      PROMPT                 ASKS THE USER WHAT MONTH,DAY,YEAR THE SCANS ARE
                               TO BE PERFORMED
08      STO 02                 STORE THAT NUMBER IN REGISTER 02
09      "START H.MS?"
10      PROMPT                 ASK THE USER WHAT HOUR,MIN,SECOND THE SCANS ARE
                               TO START (TIME FORMAT IS 2400 hr)
11      "^AUTOS"               ENTERY POINT FOR SCANS
12      XYZALM                 SETS ALARMS FOR DATE IN THE X REGISTER, THE TIME
                               IN THE Y REGISTER,THE SCAN INTERVAL IN THE Z
                               RESGISTER
13      HR
14      RCL Z                  RECALL THE REPEAT INTERVAL OF THE ALARM
15      HR
16      0.50
17      *
18      -
19      "LOG H.MS?"
20      PROMPT                 ASKS THE USER HOW MANY TIMES DO YOU WANT TO LOG
                               THE DATA THAT IS COLLECTED.
21      STO 03
22      HR
23      +
24      HMS                         CONVERTS THE NUMBER IN THE X REGISTER FROM
                               DECIMAL HOURS FORMAT TO HOURS, MINUTES, SECONDS
                               FORMAT
25      RCL 03
26      X<>Y
27      ENTER
28      ENTER
29      24
30      /
31      INT
32      RCL 02
33      X<>Y
34      DATE +                 CALCULATE A NEW DATE FROM THE DATE IN THE Y
                               REGISTER
35      X<>Y
36      24
37      MOD
38      "  AUTOL"
39      XYZALM
40      RTN
41      LBL "SETCLK" GO SET THE CLOCK
42      FIX 6
43      CLK 24                 SET THE CLOCK TO 24 HOUR FORMAT
44      DATE                   GO GET TODAYS DATE
45      "M.DY?"
46      ARCL X
47      PROMPT                 IS THE MONTH, DAY,YEAR CORRECT?
48      SETDATE                SET THE CLOCK TO THE DATE IN THE X REGISTER
49      TIME
50      "H.MS?"
51      ARCL X
```

```
52        CF 22
53        PROMPT                IS THE TIME CORRECT?
54        FS? 22
55        SETIME                SET THE CLOCK TO THE TIME IN THE X REGISTER
56        RTN
57        END


PROGRAM NAME:  EINT

FUNCTION:  INTILIZE THE DATA FILE CALLED LOGGER

REGISTER USE:
          10   STATION NAME
          15   OUTPUT/INPUT PRESSURE ELEVATION DIFFERENCE
          16   INPUT PRESSURE SENSOR OFFSET
          17   INPUT PRESSURE SENSOR SLOPE
          18   OUTPUT PRESSURE SENSOR OFFSET
          19   OUTPUT PRESSURE SENSOR SLOPE
          20   TOTAL NUMBER OF FLOW METERS
          30   TOTAL NUMBER OF WATT METERS
          39   ENERGY COST
01        LBL "EPINT"           GLOBAL ENTERY LEVEL
02        "LOGGER"              NAME OF THE FILE THAT DATA WAS STORED IN
03        30
04        SF 25                 SET FLAG 25.WITH FLAG 25 SET THE CALCULATOR WILL
                                IGNORE 1 ERROR
05        CRFLD                 CREATE A DATA FILE
06        CF 25                 CLEAR FLAG 25
07        "STA:  "              SHOW THE STATION NAME
08        10                    REGISTER 10 HAS THE STATION NAME
09        XEQ A
10        "IP-A?"               WHAT IS THE WATER LEVEL OFFSET
11        16                    REGISTER WITH WATER LEVEL OFFSET
12        XEQ B
13        "IP-B?"               WHAT IS THE WATER LEVEL SLOPE
14         17                   REGISTER WITH THE WATER LEVEL SLOPE
15        XEQ B
16        "OP-A?                WATER IS THE DISCHARGE PRESSURE OFFSET
17        18                    REGISTER WITH DISCHARGE PRESSURE OFFSET
18        XEQ B
19        "OP-B?"               WHAT IS THE DISCHARGE PRESSURE SLOPE
20        19                    REGISTER WITH DISCHARGE PRESSURE SLOPE
21        XEQ B
22        "O-I ELEV?"           WHAT IS THE OUTPUT / INPUT ELEVATION FERENCE
23        15                    PUT IT IN REGISTER 15
24        XEQ B
25        "S-A "                WHAT IS THE SPEED OFFSET
26        13
27        XEQ B
28        "S-B "                WHAT IS THE SPEED SLOPE
29        14
30        XEQ B
31        "MAX EFF?"            WHAT IS THE PUMPS MAXIMUM EFFICIENCY IN CENT
32        24
33        XEQ B
34        RCL 27
35        INT
36        STO27                 PUT IT IN REGISTER 27
37        27
38        RCL 27
```

```
39        ENTER
40        ENTER
41        2
42        +
43        1000
44        /
45        +
46        STO 26
47        STO 27
48        "WM-A "              WHAT IS THE WATT METER OFFSET
49        31
50        XEQ B
51        "WM-B "              WHAT IS THE WATT METER SLOPE
52        32
53        XEQ B
54        "$/KWH?"  WHAT IS THE ENERGY COST (CALL YOUR UTILITY COMPANY FOR THE
                               CURRENT RATE)
55        39
56        XEQ B
57        "Q-A "               WHAT IS THE FLOW METER OFFSET
58        21
59        XEQ B
60        "Q-B "               WHAT IS THE FLOW METER SLOPE
61        22
62        XEQ B
63        "PA:-"
64        23
65        XEQ B
66        EMROOM               GO GET THE NUMBER OF REGISTERS AVAILABLE FOR A
                               NEW FILE
67        8
68        /
69        INT
70        8
71        *
72        STO 25
73        CLA                  CLEAR THE DISPLAY
74        ARCL 10              RECALL THE STATION NAME
75        35
76        XTOA
77        RCL 25               RECALL THE NUMBER OF REGISTERS AVAILABLE
78        SF 25                SET THE ERROR IGNORE FLAG
79        CRFLD                CREATE A DATA FILE
80        CF 25
81        XEQ "SITSTO"         GO TRANSFER THE DATA FROM MAIN MEMORY TO THE
                               EXTENDED MEMORY
82        XEQ "SETCLK"         GO SET THE CLOCK
83        XEQ "ESTLOG"         GO SET THE ALARMS
84        XEQ "SITRCL"         GO GET THE SITE CONFIGURATION DATA
85        RTN
86        END
```

PROGRAM NAME: RAMTSTO

FUNCTION: STORE DATA TO THE RAM DISK

```
01        LBL "RAMSTO"         GLOBAL ENTRY LEVEL
02        SF 25
03        0
04        STO 05
```

```
05        "MASTER"              NAME OF THE RECORD THAT CONTAINS FILE
                                NFIGURATION INFORMATION
06        SEEKR                 SEEK THE RECORD CALLED MASTER
07        "NO MASTER"           ERROR MESSAGE IF THE MASTER FILE CANNOT  LOCATED
08        FC? 25
09        GTO 99
10        LBL 05                SUBROUTINE ENTRY POINT
11        4                     PUT 4 IN THE X REGISTER
12        ST+05                 ADD THIS NUMBER TO 4 AND STORE IT IN REGISTER
13        6.009                 REGISTER BLOCKS 06 - 09
14        READRX                COPY THE DATA FILE USING REGISTERS 06-09
15        "EOF"                 ERROR MESSAGE IF THE FILE CANNOT BE FOUND
16        FC? 25
17    RTN
18        RCL 06                RECALL THE WATER LEVEL
19        RCL 10                RECALL THE STATION NAME
20        X NE Y?
21        GTO 05
22        4
23        ST-05
24        FIX 6
25        FC 55?                IS THE PRINTER HOOKED UP
26        GTO 10
27        ADV                   ADVANCE THE PRINTER PAPER
28        "STA: "
29        ARCL 10
30        PRA                   PRINT THE STATION NAME IN THE ALPHA REGISTER
31        "DATE:   "
32        DATE
33        ARCL X
34        PRA                   PRINT TODAYS DATE
35        ADV
36        LBL 10
37        CLA                   CLEAR THE DISPLAY
38        ARCL 10
39        35
40        XTOA
41        RCL 11                RECALL THE READ POINTER
42        RCL 25                RECALL THE EXTENDED MEMORY FILE SIZE
43        MOD                   CALCULATES THE REMAINDER OF Y DIVIDED BY X
44        SEEKPTA               SET THE POINTER TO THE NUMBER IN THE X REGISTER
45        50.057                REGISTER BLOCK THAT CONTAINS THE DATA
46        GETRX                 COPY FILES 50 TO 57
47        FS? 55                IS THE PIRNTER CONNECTED
48        PRREGX                PRINT REGISTERS 50 TO 57
49        CLA                   CLEAR THE DISPLAY
50        ARCL 10
51        RCL 08
52        SEEKR                 POSTION THE WRITE POINTER TO THE CORRECT FILE
53        50.057
54        WRTRX                 COPY REGISTERS 50 TO 57 TO THE RAM DISK
55        8
56        ST +08
57        ST +11
58        RCL 12                RECALL THE EXTENDED MEMORY WRITE POINTER
59        RCL 11                RECALL THE EXTENDED MEMORY READ POINTER
60        X<Y?                  CHECK TO SEE IF ALL THE DATA HAS BEEN TRANSFERED
61        GTO 10
62        ISG 00
63        GTO 10
64        RCL 05
```

```
65        "MASTER"
66        SEEKR
67        6.009
68        WRTRX
69        XEQ "SITSTO"
70        CF 25
71        "DONE"
72        RTN
73        END
```

PROGRAM NAME: SPEED (SPD)

FUNCTION: CALCULATES THE SPEED OF THE PUMP (R.P.M)

```
01        LBL "SPD"
02        0
03        STO 08                CLEAR REGISTER 08
04        RCL 13                RECALL SPEED OFFSET
05        RCL 14                RECALL SPEED SLOPE
06        X=Y?
07        RTN
08        55
09        ASCAN                 GO SCAN THE ANALOG CHANNELS
10        RCL 14                RECALL THE SLOPE
11        *
12        RCL 13                RECALL THE OFFSET
13        +
14        STO 08                STORE IT IN REGISTER 08
15        CLA
16        ARCL 08
17        "R.P.M:   "
18        RTN
19        END
```

PROGRAM NAME: SUMIT

FUNCTION: CHECKS DATA THEN STORES IT

```
01        LBL "SUMIT"
02        RCL 04                RECALL THE INPUT HORSEPOWER
03        ST +42                ADD IT THEN STORE IT IN REGISTER 42
04        RCL 05                RECALL THE FLOW
05        ST +43                ADD IT THEN STORE IT IN REGISTER 43
06        RCL 06                RECALL THE WATER LEVEL
07        ST +44                ADD IT THEN STORE IT IN REGISTER 44
08        RCL 07                RECALL THE DISCHARGE PRESSURE
09        ST +45                ADD IT THEN STORE IT IN REGISTER 45
10        RCL 08                RECALL THE SPEED
11        ST +46                ADD IT THEN STORE IT IN REGISTER 46
12        RCL 09                RECALL THE EFFICIENCY
13        ST +47                ADD IT THEN STORE IT IN REGISTER 47
14        1                     PUT 1 IN THE REGISTER
15        ST +41                ADD 1 TO THE TOTAL OBSERVATIONS OF THE SCAN
                                PERIOD
16        RTN
17        END
```

The following is the copy of the HP41CX program, RAMDUMP, which transfers data to a PC running the BASIC program found in Appendix 3.

PROGRAM NAME: RAMDUMP

FUNCTION: USED TO TRANSFER DATA FROM THE RAM DISK TO THE P.C.

```
01          LBL "RAMDUMP"           GLOBAL ENTERY LEVEL
02          "CMTDISK1"              ADDRESS OF
03          SF 25
04          FINDID
05          "NO RAM"
06          X=0?
07          GTO 99
08          FC? 25
09          GTO 99
10          1
11          +
12          STO 04
13          SF 17
14          AUTO I/O
15          0
16          STO 05
17          LBL 00
18          "MASTER"
19          SF 25
20          RCL 05
21          SEEKR
22          FC? 25
23          GTO 15
24          006.009
25          READRX
26          FC? 25
27          GTO 15
28          CF 25
29          RCL 09
30          RCL 08
31          X<=Y?
32          GTO 12
33          RCL 07
34          9
35          +
36          .01
37          +
38          1000
39          /
40          10
41          +
42          STO 02
43          RCL 07
44          100
45          /
46          RCL 08
47          1
```

```
48          -
49          +
50          1000
51          /
52          RCL 09
53          +
54          STO 01
55          CLA
56          ARCL 06
57          SEEKR
58          XEQ 50
59          13
60          XTOA
61          FIX 0
62          CF 29
63          ARCL 07
64          13
65          XTOA
66          SCI 9
67          OUTA
68          LBL 05
69          XEQ 60
70          RCL 02
71          STO 00
72          READRX
73          XEQ 50
74          LBL 10
75          CLA
76          ARCL IND 00
77          32
78          XTOA
79          OUTA
80          ISG 00
81          GTO 10
82          CLA
83          13
84          XTOA
85          OUTA
86          ISG 01
87          GTO 05
88          CLA
89          35
90          XTOA
91          13
92          XTOA
93          OUTA
94          XEQ 60
95          RCL 08
96          STO 09
97          'MASTER'
98          RCL 05
99          SEEKR
100         006.009
101         WRTRX
```

```
102         "READY"
103         AVIEW
104         STOP
105         LBL 12
106         4
107         ST+ 05
108         GTO 00
109         LBL 15
110         CLA
111         33
112         XTOA
113         13
114         XTOA
115         OUTA
116         "DONE"
117         LBL 99
118         AVIEW
119         RTN
120         LBL 50
121         RCL 04
122         SELECT
123         MANIO
124         RTN
125         LBL 60
126         AUTOIO
127         END
```

THE FOLLOWING IS A EXAMPLE OF THE OUTPUT THAT WAS OBTAINED FROM THE PROGRAM

STA:	NWOODS	THIS IS THE STATION NAME
DATE:11.201989		IS THE DATE THAT THE DATA WAS TRANSFERED TO THE RAM DISK WHICH WAS NOVEMBER 20,1989
R50=	1,107.175234	REGISTER 50 CONTAINS THE DECIMAL EQUIVALENT OF THE CURRENT DATE AND TIME
R51=	16.00	SHOWS THE NUMBER OF SUCCESSFUL SCANS DURING A SCAN PERIOD. (i.e If the pump was on)
R52=	78.796363	REGISTER 52 CONTAINS THE INPUT HORSEPOWER
R53=	0.720638	REGISTER 53 CONTAINS THE CURRENT FLOW IN CFS
R54=	-12.024179	REGISTER 54 CONTAINS THE WATER LEVEL THE MINUS SIGN INDICATES THE DEPTH FROM THE PUMP BASE TO THE WATER LEVEL
R55=	199.940469	REGISTER 55 CONTAINS THE CURRENT DISCHARGE PRESSURE
R56=	1,324.951172	REGISTER 56 CONTAINS THE PUMP SPEED IN RPM'S
R57=	20.467971	REGISTER 57 CONTAINS THE AVERAGE PUMP EFFICIENCY FOR THE SCAN PERIOD

APPENDIX 3

The following are the **MICROSOFT QUICK BASIC** programs that you will need to put on your PC in order to transfer data from the RAMDISK to your PC, also will allow you to print the data from your PC to your printer.

```
01  DECLARE SUB READMAST (STANUM!)
02  DECLARE SUB HEADER (N!, D$, X$, PAGE!, T$)
03  DECLARE SUB PUMPHEAD (D$, X$)
04  DECLARE SUB LEVELHEAD (D$, X$)
05  DECLARE SUB XCHNGHEAD (D$, X$)
06  DECLARE SUB EFFMON (D$, T$, X$)
07  ' PROGRAM HPOUTPUT.BAS
08  '
09  ' This program is for output of data gathered by HPINPUT.BAS from
10  ' HP41CX monitors.  The program, for each data file stored, identifies
11  ' type of file; prints a header for the file; and then prints the data
12  ' in tabular fashion.
13  '
14  ' Microsoft QuickBASIC v. 2.0
15  ' IBM Personal Computer
16  '
17  OPTION BASE 1
18
19  ' Arrays to Store Master database file - GLOBAL
20
21  DIM SHARED STA$(40), LENR(40), DESCRP$(40), TYPE$(40), CODE$(40)
22
23  ' Variables for temporary storage of data
24
25  DEFDBL Z
26  DIM ZNUM(13)
27
28  ' Open output device
29
30  OUTDEV$ = "LPT1:"
31  OPEN OUTDEV$ FOR OUTPUT AS #5
32  CALL READMAST(STANUM)
33  PICK:
34    LOCATE 20, 10
35    PRINT "                                                  "
36    LOCATE 20, 10
37    INPUT "SELECT STATION TO PRINT: ", P$
38    FOR N = 1 TO STANUM
39          IF (P$ = STA$(N)) GOTO PRTIT:
40    NEXT N
41    LOCATE 20, 10
42    INPUT "NOT FOUND, DO YOU WISH TO QUIT: ", P$
43    IF (P$ = "YES") GOTO QUIT:
44    GOTO PICK:
45
46  PRTIT:
47    LOCATE 20, 10
48    PRINT "Printing Station "; N; ":  " + STA$(N)
```

```
 49    OPEN STA$(N) + ".DAT" FOR INPUT AS #1
 50    PAGE = 0
 51  NEWPAGE:
 52    PAGE = PAGE + 1
 53    CALL HEADER(N, D$, X$, PAGE, T$)
 54    I = 0
 55    WHILE (I < 45) AND (NOT EOF(1))
 56          I = I + 1
 57          INPUT #1, MM, DD, YY
 58          PRINT #5, USING D$; MM; DD; (YY MOD 100);
 59          IF (NOT (T$ = "NO")) THEN
 60                INPUT #1, TT
 61                PRINT #5, USING T$; TT;
 62          END IF
 63          IF LENR(N) = 14 THEN
 64                FOR J = 1 TO 8
 65                      INPUT #1, ZNUM(J)
 66                      IF (ZNUM(J) > 8.999999E+09) THEN
 67                            PRINT #5, USING X$; 0!;
 68                      ELSE
 69                            PRINT #5, USING X$; ZNUM(J);
 70                      END IF
 71                NEXT J
 72                PRINT #5, : PRINT #5, SPC(38);
 73                I = I + 1
 74                FOR J = 9 TO (LENR(N) - 1)
 75                      INPUT #1, ZNUM(J)
 76                      PRINT #5, USING X$; ZNUM(J);
 77                NEXT J
 78          ELSE
 79                FOR J = 1 TO (LENR(N) - 1)
 80                      INPUT #1, ZNUM(J)
 81                      PRINT #5, USING X$; ZNUM(J);
 82                NEXT J
 83          END IF
 84          PRINT #5,
 85    WEND
 86    PRINT #5, CHR$(12);
 87    IF NOT EOF(1) GOTO NEWPAGE
 88    CLOSE #1
 89    GOTO PICK:
 90  QUIT:
 91    CLS
 92  END
 93
 94  SUB EFFMON (D$, T$, X$) STATIC
 95  PRINT #5, "                          <<<<<<<<<<<<<<<<<<<< AVERAGE 101>>>>>>>>>>>>>>>>>>>"
 96  PRINT #5, "                     Number   Input           Water  Output    103Pump  Energy"
 97  PRINT #5, "       Date   Time      of  Energy    Flow    Level    Head    105Speed     Eff"
 98  PRINT #5, "                     Scans    (hp)   (cfs)    (ft)    (ft)    107(rpm)     (%)"
 99  PRINT #5, "     ------------------------------------------------------------109-----------"
100  D$ = "    ##/##/## "
101  T$ = " #### "
102  X$ = "####.## "
```

```
103
104 END SUB
105
106 SUB HEADER (N, D$, X$, PAGE, T$) STATIC
107
108 ' This subroutine uses the file record length RLEN(N), and the station
109 ' code (CDE) to determine which table header and output format is to
110 ' be used.  At the same time, data for the station is printed: name
111 ' station type, description, and date.
112
113 PRINT #5, : PRINT #5,
114 PRINT #5, "     University of Idaho"
115 PRINT #5, "     Diversion/Pump Station Monitor/Efficiency Project"
116 PRINT #5,
117 PRINT #5, "     print date:  " + DATE$ + "     page:  "; PAGE
118 PRINT #5,
119 PRINT #5, "      Name:  " + STA$(N) + "     type:  " + TYPE$(N)
120 PRINT #5, "             " + DESCRP$(N)
121 PRINT #5,
122
123 T$ = "NO"
124 IF LENR(N) = 14 THEN
125   CALL PUMPHEAD(D$, X$)
126 ELSEIF LENR(N) = 8 THEN
127   CALL EFFMON(D$, T$, X$)
128 ELSEIF CODE$(N) = "L" THEN
129   CALL LEVELHEAD(D$, X$)
130 ELSE
131   CALL XCHNGHEAD(D$, X$)
132 END IF
133
134 PRINT #5,
135
136 END SUB
137
138 SUB LEVELHEAD (D$, X$) STATIC
139
140 'This prints the column headers for level Stations
141 ' and chooses the output format for printing.
142 '
143
144 PRINT #5, "          ------------------------------------------------"
145 PRINT #5, "                     --- Level (ft) ---  ---- Flow (cfs) ----"
146 PRINT #5, "          Date       min   mean   max    min    mean   max"
147 PRINT #5, "          ------------------------------------------------"
148 PRINT #5,
149
150 D$ = "       ##/##/##   "
151 X$ = "###.## "
152
153 END SUB
154
155 SUB PUMPHEAD (D$, X$) STATIC
156
```

```
157 PRINT #5, "    ------------------------------------------------------------------------"
158 PRINT #5, "                Min.    Mean    Max. Max/Min   Elec.          Input  Output"
159 PRINT #5, "      Date      Flow    Flow    Flow    Eff.   Power   Flow    Head    Head"
160 PRINT #5, "               (cfs)   (cfs)   (cfs)    (%)    (hp)   (cfs)   (ft)    (ft)"
161 PRINT #5, "    ------------------------------------------------------------------------"
162 D$ = "    ##/##/## "
163 X$ = "####.## "
164
165 END SUB
166
167 SUB READMAST (STANUM) STATIC
168
169 ' Open and Read Master Storage file if it exists
170
171 STANUM = 0
172
173    OPEN "MASTER.DAT" FOR INPUT AS #1
174    CLS
175    PRINT
176    PRINT "             UNIVERSITY OF IDAHO PUMP MONITOR SYSTEM": PRINT
177    PRINT "                          STATIONS ON FILE": PRINT
178    PRINT " Sta.  RL Description                    Type"
179    WHILE NOT EOF(1)
180
181     STANUM = STANUM + 1
182     LINE INPUT #1, X$
183     STA$(STANUM) = LEFT$(X$, 6):     'STA is the 6 letter name
184     XXX$ = MID$(X$, 7, 2)
185     LENR(STANUM) = VAL(XXX$):          'RLEN is the data/record
186     DESCRP$(STANUM) = MID$(X$, 9, 30): 'DESCRP is a short description of the site
187     TYPE$(STANUM) = MID$(X$, 39, 15):  'TYPE is the type of station in words
188     CODE$(STANUM) = RIGHT$(X$, 1):     'CODE is the type of station in code
189                              '  L = level, P = pump, Q = pump - flow only, X = exchange
190
191     X$ = "\    \ ## \                          \ \            \ !"
192     PRINT USING X$; STA$(STANUM); LENR(STANUM); DESCRP$(STANUM); TYPE$(STANUM); CODE$(STANUM)
193    WEND
194 CLOSE #1
195 END SUB
196
197 SUB XCHNGHEAD (D$, X$) STATIC
198
199 PRINT #5, "    ------------------------------------------------------------"
200 PRINT #5, "                        Total    Percent     Volume      Last     Last"
201 PRINT #5, "              Times     Times     Times      for day     Time     Time"
202 PRINT #5, "      Date     ON      Checked     ON       (cfs-day)     ON      OFF"
203 PRINT #5, "    ------------------------------------------------------------"
204 PRINT #5,
205
206 D$ = "    ##/##/## "
207 X$ = "####.#### "
208
209 END SUB
210
```

```
211
212 DECLARE SUB SEPERTIME (Z#, MM!, DD!, TT!)
213 DECLARE SUB FIND (NAME$, STANUM!, I!)
214 DECLARE SUB NEWSTA (NAME$, TYP!, N!)
215 DECLARE SUB TRANSLAT (REC$, ZNUM#(), KIND!)
216 DECLARE SUB SEPERDATE (Z#, MM!, DD!, YYYY!)
217 ' PROGRAM HPINPUT
218 '
219 ' This is the Input Routine for retrieving data from the HP
220 ' Ramdisk.  Data is read in, station by station, from the
221 ' RAMDISK.  The first line is the station name, the second
222 ' data per record.  Based upon the data per record, the data
223 ' records are transformed from characters strings into
224 ' individual numbers.  These numbers are stored in a database
225 ' consisting of a Master file and a file for each station which
226 ' has been dumped.
227 '
228 '  William A. Perkins
229 '  University of Idaho
230 '  Diversion and pump station monitor project July 1987
231 '  Microsoft QuickBASIC v. 2.0
232 '  IBM Personal Computer w/ Asynchronous Port
233
234
235    OPTION BASE 1
236
237 ' Arrays for master data file - Available to all routines
238 DIM SHARED STA$(20), DESCRP$(20), TYPE$(20), LENR(20), CODE$(20)
239
240 ' Arrays for reading in data
241 DEFDBL Z
242 DIM LIN$(180), ZNUM(14):  ' LIN = record  ZNUM = data
243
244 '       ****    FUNCTIONS DECLARED   ****
245 DEF FNRDCOM1$
246 ' This reads a line of information from the COM port one
247 ' character at a time.  The COM port must be previously
248 ' assigned as #1.  Characters with ASCII codes above 90
249 ' decimal are ignored.
250
251 X$ = ""
252 BEGLOOP:
253    CH$ = INPUT$(1, #1): ' Read one character
254   IF ASC(CH$) > 90 THEN
255         GOTO BEGLOOP:          'Character is not readable, read another
256   ELSE
257         IF (ASC(CH$) = 13) THEN
258               GOTO ENDLOOP:      'Last char of line is a <CR> - finished
259         ELSE
260               X$ = X$ + CH$:     'Character OK, add to string
261               GOTO BEGLOOP:
262         END IF
263   END IF
264 ENDLOOP:
```

```
265  FNRDCOM1$ = X$
266  END DEF
267  '           ****   END FUNCTION DECLARATION ****
268
269  ' Port for the computer
270  PORT$ = "COM1:"
271  ' Display some Introductory Remarks
272  CLS
273  PRINT : PRINT
274  PRINT "                  HP Monitoring Station Data Transfer"
275  PRINT
276  PRINT "   This is the input routine for transferring data, gathered from the"
277  PRINT "   various stations, from the HP RAMDISK to computer storage.  You will"
278  PRINT "   need the following equipment:"
279  PRINT
280  PRINT "            *  the office HP 41CX calculator"
281  PRINT "            *  an HP-IL module"
282  PRINT "            *  the CMT RAMDISK where station data is"
283  PRINT "                stored"
284  PRINT
285  PRINT "   Be sure all equipment is hooked up properly, and the RAMDISK's RS -232"
286  PRINT "   port is connect to the proper port on the computer.  The active port"
287  PRINT "   here is " + PORT$ + "."
288  PRINT
289  PRINT "Press any key to continue, or Q to quit...";
290  xxx$ = INPUT$(1)
291  IF xxx$ = "Q" OR xxx$ = "q" GOTO ABORT
292
293  ' Open and Read Master Storage file if it exists
294  CLS : PRINT : PRINT "   Reading Storage Data..."
295  STANUM = 0
296  ON ERROR GOTO NOMASTER: 'skip reading master if not found
297  OPEN "MASTER.DAT" FOR INPUT AS #1
298  ON ERROR GOTO 0
299  CLS : PRINT : PRINT "                  UNIVERSITY OF IDAHO HP41 MONITOR SYSTEM"
300  PRINT "                                STATIONS ON FILE": PRINT
301  PRINT " Sta.  RL Descrition                              Type"
302  WHILE NOT EOF(1)    'Read records from Master and list on screen.
303    STANUM = STANUM + 1
304    LINE INPUT #1, X$
305    STA$(STANUM) = LEFT$(X$, 6):     'STA is the 6 letter name
306    xxx$ = MID$(X$, 7, 2)
307    LENR(STANUM) = VAL(xxx$):          'RLEN is the data/record
308    DESCRP$(STANUM) = MID$(X$, 9, 30): 'DESCRP is a short description of the site
309    TYPE$(STANUM) = MID$(X$, 39, 15):  'TYPE is the type of station in words
310    CODE$(STANUM) = RIGHT$(X$, 1):     'CODE is the type of station in code
311    '  L = level, P = pump, Q = pump - flow only, X = exchange  E = Efficiency
312    X$ = "\    \ ## \                          \ \            \ !"
313    PRINT USING X$; STA$(STANUM); LENR(STANUM); DESCRP$(STANUM); TYPE$(STANUM); CODE$(STANUM)
314  WEND
315  CLOSE #1
316
317  NOMASTER: ' RESUME here if Master file not found
318
```

```
319 PRINT "press any key to continue..."
320 xxx$ = INPUT$(1)
321 IF (xxx$ = "Q" OR xxx$ = "q") GOTO ABORT
322 ' Set COM port to listen
323 PORT$ = PORT$ + "9600,N,8,1,CS0,DS0,CD0,ASC"
324 OPEN PORT$ FOR INPUT AS #1
325
326 ' Instructions for operation of calculator during dump.
327 CLS
328 PRINT
329 PRINT "          Ready to read data from HP RAMDISK ..."
330 PRINT
331 PRINT "          Start the calculator transfer test dump program by keying in:"
332 PRINT
333 PRINT "                  [XEQ] [ALPHA] RAMDUMP [ALPHA]"
334 PRINT
335
336 STALOOP:  ' Loop for input of each station. Read in data from HP
337   LN = 0
338 RECLOOP:      '  Loop to read several records for a station
339         LN = LN + 1
340         LIN$(LN) = FNRDCOM1$
341         LOCATE 10, 5
342         PRINT USING "Recieving Data...   ## LINES READ"; LN
343         LOCATE 15, 1
344         PRINT LIN$(LN)
345         IF INSTR("!", LIN$(LN)) <> 0 GOTO ALLDONE: ' End of data being sent.
346         IF INSTR("#", LIN$(LN)) = 0 GOTO RECLOOP: ' End of one station data.
347   LN = LN - 1
348   STATION$ = LIN$(1)              '    Decipher and store data records.
349   TYPENO = VAL(LIN$(2))
350   CALL FIND((STATION$), (STANUM), N):  'See if this station has a file.
351   LOCATE 12, 5
352   IF (N <> 0) THEN ' The station has been dumped before.
353         PRINT "Station:  "; STA$(N); SPC(2); LENR(N); " data per record"
354         PRINT SPC(7); TYPE$(N)
355         PRINT SPC(7); DESCRP$(N)
356         OPEN STA$(N) + ".DAT" FOR APPEND AS #2
357   ELSE            ' The station has not been previously dumped.
358         PRINT "Station:  "; STATION$; SPC(2); TYPENO; "data per record"
359         PRINT CHR$(7); "** NEW STATION **"
360         CALL NEWSTA((STATION$), (TYPENO), STANUM)
361         N = STANUM
362         OPEN STA$(N) + ".DAT" FOR OUTPUT AS #2
363   END IF
364
365    '  Loop to store transform and store data
366   IF (CODE$(N) = "E") THEN
367         INPUT "Year data collected: ", YEAR
368   END IF
369   FOR J = 3 TO LN
370         LOCATE 20, 5
371         CALL TRANSLAT(LIN$(J), ZNUM(), (LENR(N)))
372         IF (CODE$(N) = "E") THEN
```

```
373                 CALL SEPERTIME(ZNUM(1), MON, DAY, TT)
374                 PRINT USING "Storing...   DAY = ##/## ####"; MON; DAY; TT
375                 PRINT #2, USING "## ## #### ####"; MON; DAY; YEAR; TT;
376           ELSE
377                 CALL SEPERDATE(ZNUM(1), MON, DAY, YEAR)
378                 PRINT USING "Storing...   DATE = ##/##/####"; MON; DAY; YEAR
379                 PRINT #2, USING "## ## ####"; MON; DAY; YEAR;
380           END IF
381           FOR K = 2 TO TYPENO
382                 PRINT #2, USING " #.########^^^^"; ZNUM(K);
383           NEXT K
384           PRINT #2,
385     NEXT J
386     CLOSE #2
387
388     ' Instructions for operation of calculator
389     CLS
390     PRINT
391     PRINT
392     PRINT "     Ready for Next Station..."
393     PRINT
394     PRINT "         When calculator displays 'READY',"
395     PRINT "         Press the calculator's RUN/STOP (R/S) key"
396
397     GOTO STALOOP:  '  Return to beginning of station loop.
398
399   ALLDONE: '  Exit from station loop to here to finish up
400   CLOSE #1:  ' Close COM port
401
402   '  Resave Master file
403   OPEN "MASTER.DAT" FOR OUTPUT AS #1
404   FOR I = 1 TO STANUM
405     X$ = "\    \##\                              \\            \!"
406     PRINT #1, USING X$; STA$(I); LENR(I); DESCRP$(I); TYPE$(I); CODE$(I)
407   NEXT I
408   CLOSE #1
409   ABORT:
410   END
411
412   SUB FIND (NAME$, STANUM, I) STATIC
413
414   ' This finds the station name NAME$ in the arrays which represent
415   ' the master storage file.  The number in the array is returned
416   ' (I), if the name is found, otherwise a zero is returned.
417   SHARED STA$()
418   FOUND = 0
419   I = 1
420   WHILE (FOUND = 0) AND (I <= STANUM)
421      IF NAME$ = LEFT$(STA$(I), LEN(NAME$)) THEN
422       FOUND = 1
423      ELSE
424       I = I + 1
425      END IF
426   WEND
```

```
427  IF I > STANUM THEN I = 0
428  END SUB
429
430
431  SUB NEWSTA (NAME$, TYP, N) STATIC
432
433  ' This adds a new station to the master arrays.  The user is asked
434  ' for a description of the station (30 char), and may be asked to input a
435  ' station type.
436  SHARED STA$(), TYPE$(), LENR(), DESCRP$(), CODE$()
437  N = N + 1
438  STA$(N) = NAME$
439  LENR(N) = TYP
440  INPUT "Station Description:  "; DESCRP$(N)
441  IF LENR(N) = 4 THEN
442    TYPE$(N) = "DIVR FLOW      "
443    CODE$(N) = "Q"
444  ELSEIF LENR(N) = 14 THEN
445    TYPE$(N) = "DIVR FLOW/EFF. "
446    CODE$(N) = "P"
447  ELSEIF LENR(N) = 7 THEN
448    PRINT "Type number is 7 - input code for station "
449    PRINT "( X = exchange, L = level):";
450    CODE$(N) = INPUT$(1)
451    IF CODE$(N) = "X" THEN
452          TYPE$(N) = "EXCHANGE WELL"
453    ELSE
454          TYPE$(N) = "LEVEL MONITOR"
455    END IF
456  ELSEIF LENR(N) = 8 THEN
457    TYPE$(N) = "PUMP EFFICIENCY"
458    CODE$(N) = "E"
459  END IF
460  PRINT "Station Type:  "; TYPE$(N)
461  END SUB
462
463  SUB SEPERDATE (Z, MM, DD, YYYY) STATIC
464
465  ' This subprogram takes the number passed as Z, in HP date format,
466  ' mm.ddyyyy, and translates it into 3 numbers MM, DD, YYYY.
467  ' This makes the date easier to read.
468     MM = INT(Z)
469     DD = INT((100 * Z) MOD 100)
470     YYYY = (10000 * Z) MOD 1000000
471  END SUB
472
473  SUB SEPERTIME (Z, MM, DD, TT) STATIC
474
475  ' This subprogram takes the number passed as Z, in HP date format,
476  ' MMDD.TTTTT, and translates it into 3 numbers MM, DD, TT.
477  ' This makes the date easier to read.
478
479     MM = INT(Z / 100)
480     DD = Z MOD 100
```

```
481     TT = (10000 * Z) MOD 10000
482
483  END SUB
484
485
486  SUB TRANSLAT (REC$, ZNUM(), KIND) STATIC
487
488  ' This subprogram sorts a data line out into the proper number
489  ' of data.
490
491     FOR J = 1 TO KIND
492
493      LOCN = INSTR(REC$, " ")
494      xxx$ = LEFT$(REC$, LOCN)
495      ZNUM(J) = VAL(xxx$)
496      REC$ = RIGHT$(REC$, LEN(REC$) - LOCN)
497     NEXT J
498  END SUB
```