Woods Hole Oceanographic Institution, USA

WHOI-90-03

Copy 2

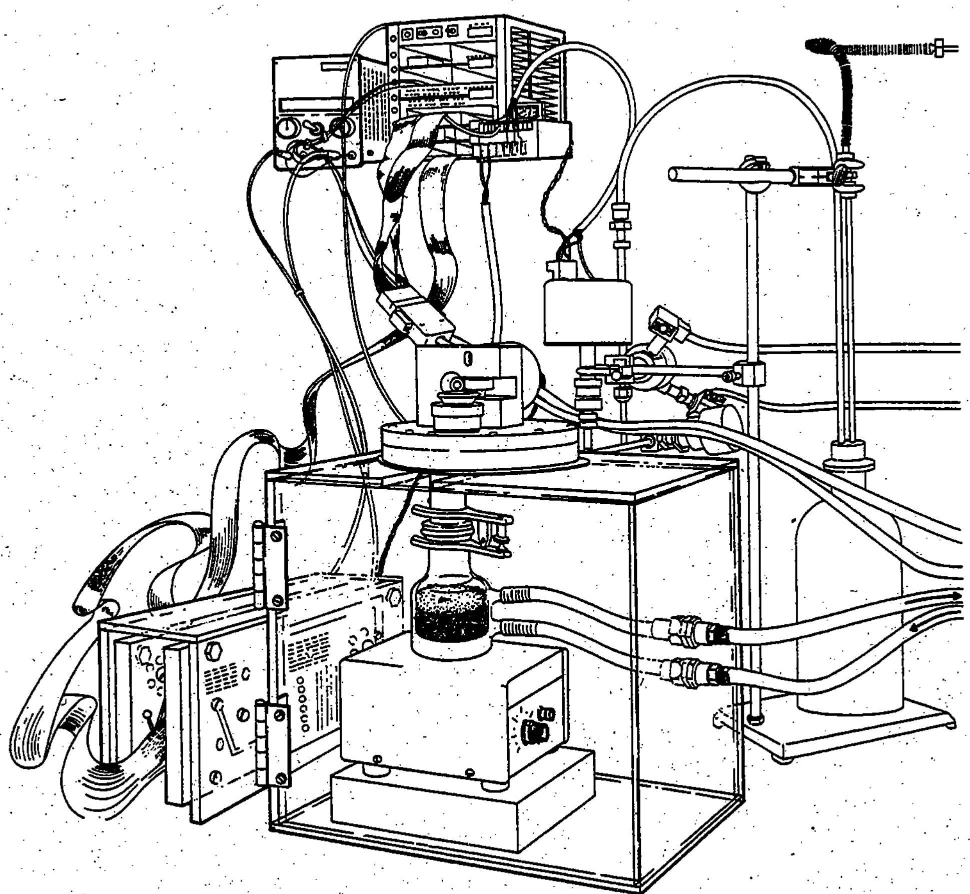
Automated System to Measure the Carbonate Concentration of Sediments

by

Dorinda R. Ostermann, Darrell Karbott and W.B. Curry

February 1990

Technical Report



WHOI-90-03

Automated System to Measure the Carbonate Concentration of Sediments

by

Dorinda R. Ostermann, Darrell Karbott and W.B. Curry

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

February 1990

Technical Report

Funding was provided by the National Science Foundation through Grant Nos. OCE 85-11014 and OCE 88-13307.

Approved for Distribution:

David A. Ross, Chairman
Department of Geology & Geophysics

Contents

1 INTRODUCTION

We have developed a computer controlled system to measure the calcium carbonate content of sediment samples. A menu driven program controls the analysis of each sample. The system first communicates with a Mettler digital balance to record the weights of the 40 samples which must be loaded into each run. The sample boats are next loaded into the sample carousel which is then sealed from the atmosphere. The system is first pumped down to a vacuum of 0.04 torr. The valve to the pump closes and the stepping motor turns the carousel, moving a sample boat over the delivery slot and dropping the sample into 80°C 100% phosphoric acid under vigorous spinning action. During the reaction, carbonate is evolved into H_2O and CO_2 and the resulting pressure change within the closed system is measured by a pressure transducer and recorded into memory next to the sample identification and sample weight. The system is pumped once again to 0.04 torr and the process continues until all 40 samples have been analyzed. The data can then be uploaded and converted to percent carbonate values using a regression line produced from multiple analyses of varying weights of a 100% carbonate standard. Precision of the system, based upon 120 replicate analysis ranges from 0.49% to 0.88%.

2 HARDWARE

We set out to develop an easy-to-use automated carbonate system which was inexpensive and microprocessor controlled. We based the system on the STD bus because of its modular control-oriented system design and because of the wide availability of plugin cards for specific control applications. With a minimal amount of rejumpering (see Appendix 1), the boards were snapped into the card cage, and connected to the balance, terminal, stepping motor and pressure transducer. The automated carbonate system (ACS) consists of the following major components, and is diagrammed in Figure 1.

2.1 Microprocessor

The **VERSALOGIC©** VL-7806 Z80 microprocessor is used in this application. The STD bus compatible board features a flexible memory mapping option which allows a mixture of memory devices to be used in the system. In this system we use a 64K memory expansion board in addition to the memory contained on the Z80 board. Appendix 2 details the memory map of the Z80 processor as configured in this system. The Z80 board also features two RS232 serial I/O ports. We use port B to communicate with the **METTLER** AE100 balance through a bi-directional interface. Port A is connected to a **NEC** APC II computer which is used as a terminal for the Z80 processor and to upload the data from the carbonate analyzer. A *C4 BASIC©* 8K ROM is plugged into socket M0 on the VL-7806 processor card. The *C4 BASIC* language which runs the automation is oriented to process control tasks and includes special statememts for I/O device interface. It is designed to run programs directly from ROM storage and has an operating system, *NOVOS*, which utilizes RAM nonvolatile memory. This allows development, testing, backup, copying and transferring of *C4 BASIC* programs in a totally diskless system.

2.2 Memory Expansion

The large size of the automation program made it necessary to expand the memory of the system using a **VERSALOGIC** VL-7709 memory card. The board provides 16-bit addressing for use with the Z80 processor. The system as configured uses a total of 48K of RAM, three 8K chips on the Z80 board and three 8K chips on the memory board. This 48K of memory may be allocated for storage area and development RAM in any proportion using the system SETUP command (see Appendix 6). However, the sum of the number of Kbytes for storage and development RAM *must not* exceed 47K. This reserves 1K at the top of memory (&DC01-&DCFF) for critical machine language subroutines.

2.3 Relay Output

Solenoid actuators for the pneumatic valves are controlled by a **VERSALOGIC** IPI-2 eight channel relay output card. The individual relays are powered by a 28 VDC source and the channels we use are activated through address &96. Circuitry is provided to drive eight LED indicators at the edge of the board which display the current on or off state of each channel.

2.4 Pressure Transducer

A 0-100 torr **BARATRON©** absolute pressure transducer is used as the system pressure sensor. We chose this particular transducer because it is capable of withstanding high over-pressure conditions and it is highly resistant to corrosive environments. We connected the transducer and the analog input board through a 10 K reostat (see Appendix 3A). The reostat reduced the normal 10 volt output range of the transducer to 5 volts, doubling the resolution of the transducer. Careful attention was paid to limiting the internal volume of the reaction vessel and the lengths and internal diameters of the stainless steel tubing, minimizing the overall volume of the system and maximizing the pressure transducer output (see Appendix 3B). A microvoltmeter connected to the transducer externally monitors the pressure within the system.

2.5 Analog Input

The **VERSALOGIC** STD AIN-1 is an eight channel integrating analog input board compatible with the STD bus. This board uses an integrating conversion technique to perform the analog to digital conversion. The integrating method measures the actual input signal during the conversion giving a much higher accuracy in changing or moving input signals such as found in our system. The integrating conversion is also very noise tolerant. These advantages are a trade off to slower conversion times but because our system is only monitoring one signal during a 5 to 20 minute reaction time, the slower conversion time does not limit this system's efficiency. The analog input board is jumpered to a 5 volt input range. This lower voltage range provides twice the resolution for each measurment (2.5 mv versus

5 mv in the 10 mv range) without affecting stability or conversion time. The one part in 4000 (12 bit) resolution provides a digital pressure unit equivalent to .025 torr. We chose to use the differential input mode for signals that are not referenced to a ground point but are simply a voltage difference between two input wires. This mode is especially beneficial in electrically noisy environments. We communicate with this board via address &70.

2.6 Stepping Motor Driver/Controller

The **Matrix** Dual Stepping Controller (DSC) and Unipolar Stepper Driver (USD) provide microprocessor control of unipolar stepping motors. A 26-position ribbon cable connects the DSC in the card cage to the USD. The USD converts the DSC output signals to the voltage and current levels required by the stepping motor. The USD receives logic power directly from the DSC which is initialized upon power-up via a software status/command register. These commands include direction of rotation, single versus continuous step mode and the motor speed. The 4-phase stepping motor used in this application is powered by an external 24 VDC power supply connected directly to the USD.

2.7 Sample Carousel

A 'lazy susan' designed by R.G. Fairbanks served as the basis for our carousel assembly design. It was modified to our specific application and machined out of 316 stainless steel to resist acid corrosion. A four phase stepping motor is attached to the top of the carousel assembly. The motor turns a set of reducing gears connected to the carousel shaft which is sealed from the atmosphere with two 'O' rings, allowing the shaft to rotate the sample dispenser under vacuum. The carousel makes one full revolution every 2000 steps of the stepping motor. The sample carousel holds 40 samples which may be loaded through the top assembly by removing an **ULTRA-TORR©** fitting. The top and bottom carousel assemblies are secured together with allen cap screws and are sealed from the atmosphere with an 'O' ring which fits into a groove in the bottom assembly.

2.8 Reaction Vessel

A glass reaction vessel (see Appendix 3C) is attached to the bottom of the sample carousel assembly with a pinch clamp and is sealed from the atmosphere with an 'O' ring. The 20 ml phosphoric acid (see Appendix 4) within the vessel is maintained at 80°C by a circulating water bath and a strong spinning action of a magnetic stir bar.

2.9 Vacuum Line

Any pump which can pump down to a vacuum of 10^{-4} torr is suitable for use with this system. Large internal diameter 0.25 inch copper refridgeration tubing was used between the valves and the vacuum pump to provide a more efficient initial pump down. Flexible stainless steel tubing was used between the vacuum pump and the oil trap to lessen vibration through the system. **ULTRA-TORR** fittings were used in all applications except for the attachment of the pneumatic valves to the tubing where we used swagelok fittings provided by the manufacturer. Appendix 3B details the vacuum line specifics as used in this system.

3 SOFTWARE

A complete listing of one **MICROSOFT** Fortran and four *C4 BASIC* programs can be found in Appendix 5. Each program will be individually described below.

3.1 Clear.bas

Clear.bas is a *C4 BASIC* program which clears the existing storage area so that *NOVOS* can be used to reinitialize the system. Download the program and save it in an empty slot of the directory and run it. The program will ask the user to input the starting address of the storage area. If you are unsure what the storage address is, type *NOVOS*, and the address of the storage area will be printed on the screen. In the present configuration, input '&7400'. (Remember to enter the hex symbol '&' before the address number) The system will either lock up and you will have no keyboard control or 'Enter program number' will be printed.

In either case, power the system down for 15 seconds and reset the button on the power supply card. Now the system is ready to be reconfigured following the procedure outlined in Appendix 6.

3.2 Loader.bas

Loader.bas is a *C4 BASIC* program which loads into memory, two Z80 machine language subroutines which perform I/O operations not provided for in *C4 BASIC*. The first, RDKEY, reads a single key stroke from the **NEC** console, returning the ASCII code of the key pressed via the *C4 BASIC* function USR(X). The second, METIO, reads a string of up to 16 characters from the **METTLER** digital balance through serial port B on the Z80 board, storing their ASCII codes in a buffer from which they may be read by means of the *C4 BASIC* function PEEK. Installing these two subroutines is complicated by the fact that *C4 BASIC* has no built in commands to store machine language routines in its file system or to load them into memory. *Loader.bas* was therefore written to use the *C4 BASIC* function, POKE, to poke them into memory one byte at a time starting at &DC01. Although the subroutines are less than 100 bytes long, it was necessary to reserve a full 1K of memory for the routines because that is the minimum that *C4 BASIC* configuration procedure allows. The system has already been configured to protect the addresses &DC01-&DCFF from *C4 BASIC*. However, if the system is ever reconfigured, *it is essential that these addresses be reserved.*

3.3 Getdata.bas

Getdata.bas is a *C4 BASIC* program used to make consequetive measurements of reaction time versus digital pressure on a single sample. It is useful for leak testing of the system as well as determining the necessary reaction time needed for the complete evolution of CO_2 gas from a sample. Download the program into an empty directory location. Next load a sample boat into whichever load slot is to the right of the delivery slot, close the carousel, and run the program. The user will be asked to input the numbers of paired data to be collected (80 maximum). The user will next be asked to input the interval in seconds between measurments.

Decide the time interval over which you want to collect data and plan the number of data points collected and seconds between each collection accordingly. At the program end hit <P> to print the data to the screen or <Q> to quit. If you wish to copy the data to a disk, you will need to use a screen capture program. We use *ASYNC* as described on page 11.

3.4 Percent.for

Percent.for is a **MICROSOFT** Fortran program which determines the percent carbonate for samples run on the Automated Carbonate System. At least two runs of 100% carbonate standards are first run to produce a regression line of slope and intercept for each batch of acid (see **OPERATIONS**, p. 14). The program can then be recompiled with the new slope and intercept data or interactively on the screen each time the program is used. The user will first be asked for the input filename. (By convention we always name the raw data from the carbonate system with 'filename.crb'). The slope and intercept will then be listed. If you wish to use the default values hit <RET>, otherwise insert the values you wish to use. Finally, the user will be asked to name the output file. (By convention we always name the converted data file 'samefilename.pct'). The program will crash if 'samefilename.pct' already exists. If it does, simply delete it and rerun the program.

3.5 Animat.bas

Animat.bas is the *C4 BASIC* program which controls the Automated Carbonate System. The program is almost exculsively menu driven through two menus: (1) The Main Menu and (2) Tweak the System Menu. Each of the menus is detailed below including a few 'hidden' functions which may be called at any time.

3.5.1 The Main Menu

The Main Menu
<W> to weigh the samples
<L> to load the sample boats
<R> to run samples
<P> to print the data
<I> to reinitialize
<T> to tweak the system
<Q> to quit

<W> *to weigh the samples*

This option is used to weigh out the samples into the sample boats and to enter their corresponding depth values. It has several 'hidden' functions which can be accessed through the use of control characters. Typing '^ Z' at any prompt returns the user to the main menu. If the balance drifts after it has been tared, it can be retared by typing '^ T' at the 'Add the sample to the boat and hit any key when it is ready to be weighed=>' prompt. If the balance drifts after the sample has been weighed it can be reweighed by entering '^ W' at the 'Do you wish to redo this sample? (Y/N)' prompt. Occassionally the balance will refuse to tare or weigh on the first sample only. This problem seems to occur when the balance has been turned off or has been used between carbonate runs. Abort to the main menu by typing 'A'. Turn the balance off and on several times and then reboot the system by running the Loader and Automated Carbonate System programs. Finally, reinitialize the memory.

<L> *to load samples*

This option prompts the user to load the sample boats one at a time into their proper holes in the lazy susan's carousel. The sample carousel *must* be correctly aligned at the beginning of the load sequence for proper system operation (see Appendix 7). If the lazy susan jams while loading or is at the wrong load slot:

1. Exit to the main menu by typing '^ Z'
2. Call up the 'Tweak the system menu' by typing 'T'
3. Select the 'Home the lazy susan' option by typing 'H'

4. Home the lazy susan so that the hole with the same number as the next sample you wish to load in in the load slot position. (see Appendix 7)

5. Exit to the 'Tweak the system menu' by typing 'E'

6. Exit to the main menu by typing 'E'

7. Reenter the Load option by typing 'L' An error message will be printed and this option will refuse to execute if (a) all 40 samples have not been weighed out before it is called or (b) the samples have already been run.

<R> *to run the samples*

This option reacts the individual samples and stores the pressure values in the systems internal file structure so that they may be uploaded using the '<P> print the data' option at the end of each run. The user is prompted to select one of two reaction times for analysis of individual samples based on the expected carbonate content. If the lazy susan jams while the 'Run the samples' option is executing, an error message will appear on the screen.

1. Hit the Escape key, '<Esc>'. This breaks the Animating Element. A system prompt, '*', should appear.

2. Type 'run 1' to reexecute the ACS program. The prompt 'Skip software initialization? (Y/N)', should appear.

3. Type 'Y' to skip the initialization. This preserves the systems internal file structure.

4. Type 'N' at the 'Is the lazy susan homed to slot #1? (Y/N)', prompt. The Homer menu will be displayed.

5. Home the lazy susan to correct the jam, preserving the jammed sample if possible.

6. Exit to the 'Tweak the system...' menu by typing 'E'.

7. At the prompt, enter the number of the hole in the carousel which is over the load slot, *not* the number of the next sample to be run. (see Appendix 7)

8. Type 'R' at the 'Tweak the system...' menu to resume running.

Do not run any other program or call *NOVOS* between the time you break the program by typing 'Esc' and the time you reexecute the ACS program. The data in the system's file structure (i.e. the weights of all the samples and the pressure values of the samples which have already been run) will be destroyed. The 'R' option will not execute until all the samples have been weighed and loaded.

<P> *print the data*

This option is used to print out the system's data file so that it may be uploaded to a floppy disk using *ASYNC*'s 'Get a file' function.

1. After the 40 samples has been been run and the main menu is listed, type '^ VF' to call *ASYNC*'s 'Specify file name for transfer' function. The *ASYNC* prompt 'File specification:' will appear.

2. Enter the drive and filename the data is to be saved under and hit '<return>'.

3. Type '^ VG' to call *ASYNC*'s 'Get a file' function. The message 'Port–> filename/File open, rdy to rcv' will appear on the top left corner of the screen.

4. Type 'P'. The data will scroll on the terminal one line at a time.

5. When all 40 lines of data have been printed on the terminal screen, type '^ Z'. This closes the file and the message '++File received++' will appear after the main menu is printed.

<I> *to reinitialize*

This option is used to reinitialize the system so that another batch of 40 samples can be run. The execution of this option destroys all the data in memory.

<Q> *to quit*

This option is used to exit the ACS program and enter *C4 BASIC*. *Do not* execute this option unless the data has been already uploaded. It is almost always possible to reenter the ACS program program without destroying the data by typing 'Y' at the 'Skip software initialization? (Y/N) ' prompt, but it is not worth the risk. If you must, see steps #2 and #3 for unjamming under the 'Run the samples' option.

3.5.2 Tweak the System Menu

Tweak the System Menu
<A> for atmosphere
<V> for vacuum
<W> to reweigh a sample
<R> to continue after a jam up
<C> to check the pressure in the vessel
<E> to exit to the main menu
<H> to home the lazy susan

<A> *for atmosphere*
This option opens the vessel to the atmosphere by closing the valve to the pump and opening the valve to the atmosphere.

<V> *for vacuum*
This option pumps the reaction vessel down by closing the valve to the atmosphere and opening the valve to the pump.

<W> *to reweigh a sample*
This option allows the user to reweigh a sample after all samples have been weighed. The user will be prompted for the number of the sample to be reweighed and then the normal weighing protocol is followed.

<R> *to continue after a jam up*
This option is used to continue running after the normal 'run samples' has been interrupted. Do not call this option until the lazy susan has been homed so that the carousel will drop the next desired sample into the reaction vessel. See the 'Home the lazy susan' menu above.

<C> *to check the pressure in the vessel*
This function allows the user to leak test the vessel over a selected period of time. The option closes both valves so remember to reopen the valve to the pump if you want to continue pumping on the acid after checking the pressure. An error message will be printed if this option is executed while the vessel is open to the atmosphere.

<E> *to exit to the main menu*

This option returns the user to the main menu.

<H> *to home the lazy susan*

This option calls up another menu of functions which are used to home the lazy susan.

3.5.3 Homer Menu

Homer Menu
<Right arrow> = counter-clockwise
<Left arrow> = clockwise
<Up arrow> = continuous step mode
<Down arrow> = single step mode
<S> = Smart Home Option
<E> = Exit

This menu allows the user to manipulate the lazy susan's carousel from the keyboard. The '<Right arrow>' and '<Left arrow>' keys control the direction of rotation of the lazy susan. The '<Up arrow>' and '<Down arrow>' keys switch between single step mode and continuous step mode. In single step mode, the lazy susan moves a single step in the direction of the pressed direction key. In continuous step mode, the lazy susan moves in the direction of the last pressed direction key. Pressing any key except for '<Up arrow>', '<Right arrow>' or '<Left arrow>' will cause the lazy susan to stop in continuous step mode. The carousel *must* be aligned to load slot #1 at the beginning of the load sequence (see Appendix 7).

<S> *Smart Home Option*

The smart home option asks the user for the number of the hole positioned over the load slot and automatically homes the lazy susan to load slot number 1. In order to use the smart home option, you must first use the the arrow keys to home the lazy susan to the nearest load slot. *Do not* use this option if the boats have already been loaded into the lazy susan. This option chooses the direction of rotation based upon the shortest distance to the load slot and may dump loaded boats into the acid in the process of homing the carousel.

<E> *Exit*

This option prompts the user to enter the load slot which the lazy susan has been homed to and then returns to the 'Tweak the system...' menu. This program has no encoding mechanism and thus has no way of knowing where the carousel is other than asking the user. The program does not know which load slots have boats in them and *cannot keep you from dumping boats into the acid by accident*!!

4 OPERATIONS

Once the hardware and software have been installed and are functioning correctly, the phosphoric acid and reaction vessel can be calibrated by running at least two standard runs of 100% calcium carbonate. Data from two carbonate standard runs are shown in Figure 2. The slope and intercept of the regression line produced from such standard runs are used in *percent.for*, the fortran program which converts raw data produced on the ACS to percent carbonate data.

We have found that a constant volume of acid delivered to the reaction vessel is critical to produce consistent, accurate results. Room temperature 100% phosphoric acid is quite viscous. We found that the repipette manufactured by **LABINDUSTRIES©** consistently delivers accurate volumes. (A complete listing of components for the ACS can be found in Appendix 8). We adjust the 20 ml repipet to deliver two 10 ml aliquots of acid because in the process of drawing up 20 ml, bubbles always form at the pipette tip causing inaccurate volumes. A glass-covered spin bar is added to the acid-filled reaction vessel. The reaction vessel should now be attached to the carousel assembly, heated to 80°C, and put under vacuum to begin the outgassing process. Under atmospheric conditions, the microvoltmeter should read an over-pressure voltage of ±13. Once pumpdown begins, the voltage will quickly drop to below 1 volt if no leaks are present. Within 2 minutes of pumping under vigorous spinning action, the voltage should drop below 0.10 volts. The user is ready to weigh out the 100% dried reagent grade calcium carbonate standard samples into the sample boats (see Appendix 3D for sample boat specifications).

The first boat to be weighed should be listed as 'sample 1'. If the first sample listed is 'sample 0', exit the weighing procedure and reinitialize the buffer memory by hitting the <I> under 'The Main Menu'. Then proceed with the weighing process. When producing a standard carbonate run, weigh out a range of weights from 1 mg to 20 mg. The first sample should always be large, to fully saturate the acid with CO_2 (the percent carbonate value of the first sample is always 10-15% low and is deleted). Once all 40 boats have been filled, use a finnpipette to deliver 5μl of reagent grade methanol to each sample boat. We have found that on the initial pumpdown from atmosphere to vacuum, material can be lost from the boats nearest the reaction vessel delivery slot (boats 35-5). When dried, the methanol produces a thin crust which adheres the sediment particles together. Place the 40 boats into an oven to dry. The samples are dry when they are no longer shiny (5-10 minutes at 60°C). The boats can now be loaded into the sample carousel, if load slot 1 is in the correct position (see Appendix 7).

Once the **ULTRA-TORR** fitting on the carousel assembly is secure, the system is ready to run. A choice of two reaction times will appear on the screen. We have found that our initial 5 minute reaction time was not long enough to fully evolve the CO_2 from sediment samples having low carbonate contents. To determine the optimal reaction time for sediments of varying carbonate content, we conducted an experiment using *Getdata.bas.* The results of this experiment are shown in Figure 3. In general, the evolution of carbonate to CO_2 gas is complete within one minute. However, in samples with less than 25% carbonate content, it is apparent that a much longer reaction time is necessary. Data collected from a second experiment on samples with less than 40% carbonate over a 45 minute reaction time are shown in Figure 4. These results show that a 20 minute reaction time is necessary for sediment samples with less than 25% carbonate. A small but gradual increase in percent carbonate after 20 minutes is water vapor outgassing from the acid. The precision of replicate analyses of samples run on the ACS with high, medium and low concentrations of carbonate vary between 0.49% and 0.88% as shown in Figure 5.

As the run begins, the valve to the pump opens and the pressure in the system quickly drops to .12 volts (which corresponds to 0.04 torr) and the pump valve closes. A 'stable pressure' algorithm collects a pressure value every six seconds and waits for ten consecutive values which do not change by more than .01 volts. For the first few samples it may take the computer longer than a minute to accept a background stable pressure value. By the fifth sample, however, only one minute is necessary before a stable background pressure value is reached and the next sample boat is dropped into the acid. Based upon the reaction time chosen, the system will wait for that amount of time before beginning the 'stable pressure' algorithm to collect the final pressure value. The initial background pressure is subtracted from the final stable pressure to produce the digital pressure listed in the data table.

If a sample happens to be reacting at the end of the chosen reaction time, the system will wait for the reaction to be complete before a stable pressure value is accepted and the next sample is dropped. If there is a leak in the system the pressure will never stabalize and the run will not continue. When leak is so bad that atmospheric pressure is reached, an error message will appear on the screen and the run will be terminated.

At the end of each run, the main menu will be printed on the screen. We recommend that the data from each complete run be uploaded onto a floppy disk as soon as possible. At present, the system has no battery backup protection and when the power fails, the data is lost. You can expect a typical 'high carbonate' run to take 4.5 hours while a 'low carbonate' run will take at least 14.5 hours. For this reason we run high carbonate samples during the day and the low carbonate samples overnight for maximum data output. *Do not* increase the sample weights of low carbonate samples in order to produce a greater pressure signal. The clays in low carbonate samples will turn the acid into a thick mass making the run invalid. We keep the sediment sample weight in each boat within the range of 15 to 20 mg of material for best results. If problems are encountered, refer to the 'Trouble Shooting guide in Appendix 9. For a condensed 'System Startup' procedure, see Appendix 10.

REFERENCES

Coplen, T. B., Kendall, C., and Hopple, J., 1983, Comparison of stable isotopic reference samples: Nature, v. 302, p. 236–238.

Curry, W.B., and Lohmann, G.P., 1985, Carbon deposition rates and deep water residence time in the equatorial Atlantic Ocean throughout the last 160,000 years, *in* Sundquist, E. T., and W. S. Broecker, eds., The carbonate cycle and atmospheric CO_2: natural variations, Archean to present: American Geophysical Union Monograph 32, p. 285-301.

Acknowledgements

We would like to thank E. Franks, C. Goyet, L. Keigwin, K. Miller, D. Oppo and N. Slowey for helpful criticisms of the many drafts of this manual. We credit V. Asper for his initial inspiration and technical support as we began development of the carbonate system. We also thank P. Oberlander of the W.H.O.I. Graphics Department for his beautiful illustrations of the carbonate system and it's associated parts. The development of the carbonate system was supported by NSF grants OCE85-11014 (W. B. Curry) and OCE88-13307 (W. B. Curry).

Figure 1. Automated Carbonate System Schematic.

A. 28 volt DC power supply (for stepping motor & valve solenoids)

B. Card cage including the following boards:

1. power supply (5 & 12 volt DC)
2. Z80 computer
3. Analog input
4. Memory expansion
5. Dual stepper controller
6. Relay output

C. Microvoltmeter (to display transducer pressure)

D. 15 volt DC power supply (for pressure transducer)

E. Pressure transducer

F. Pneumatic valves with attached solenoid actuator

G. Stepping motor

H. Lazy susan sample carousel

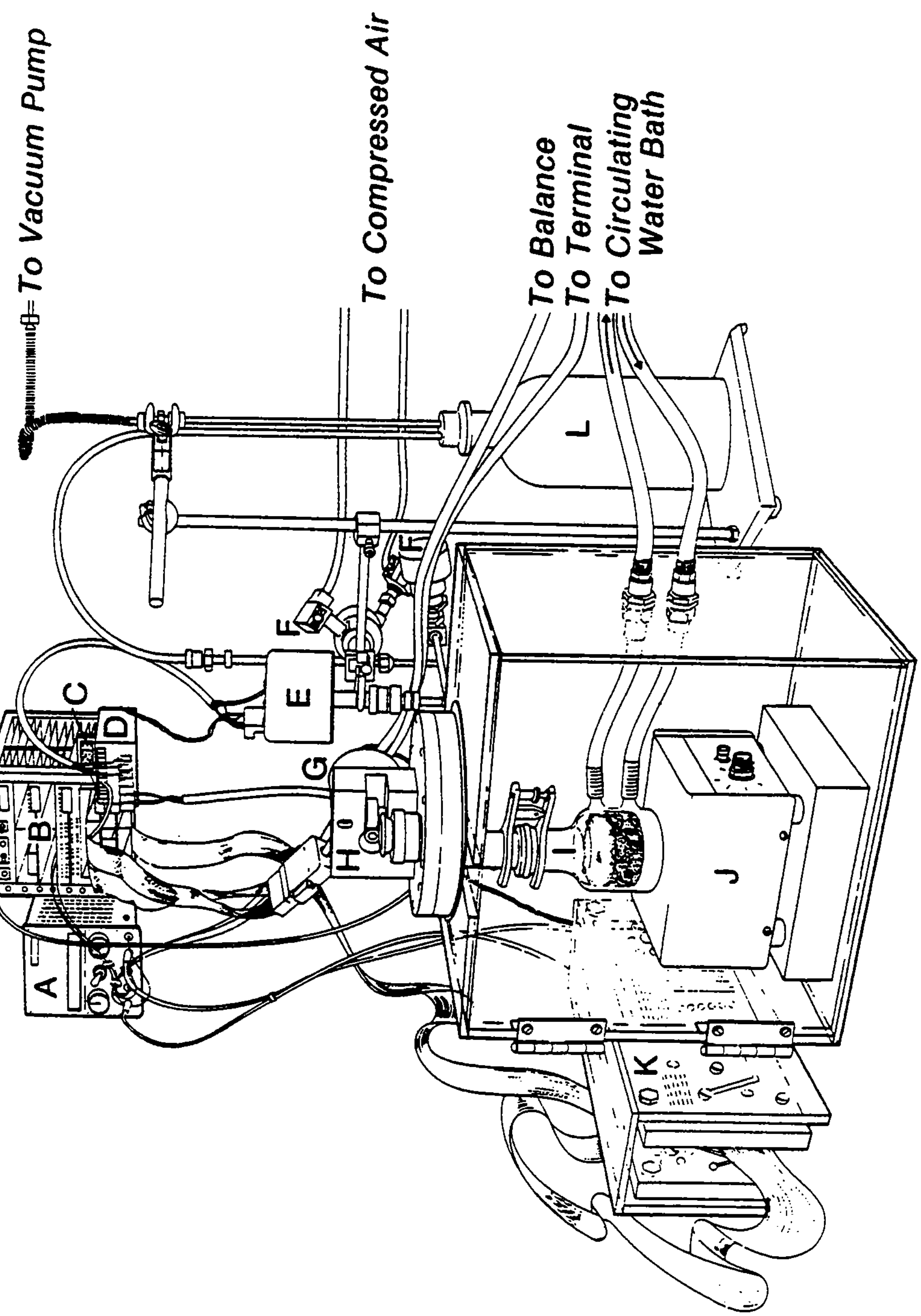
I. Water jacketed reaction vessel

J. Spin bar mixer

K. Unipolar stepper driver

L. Vacuum pump oil trap

Figure 1. Automated Carbonate System Schematic.



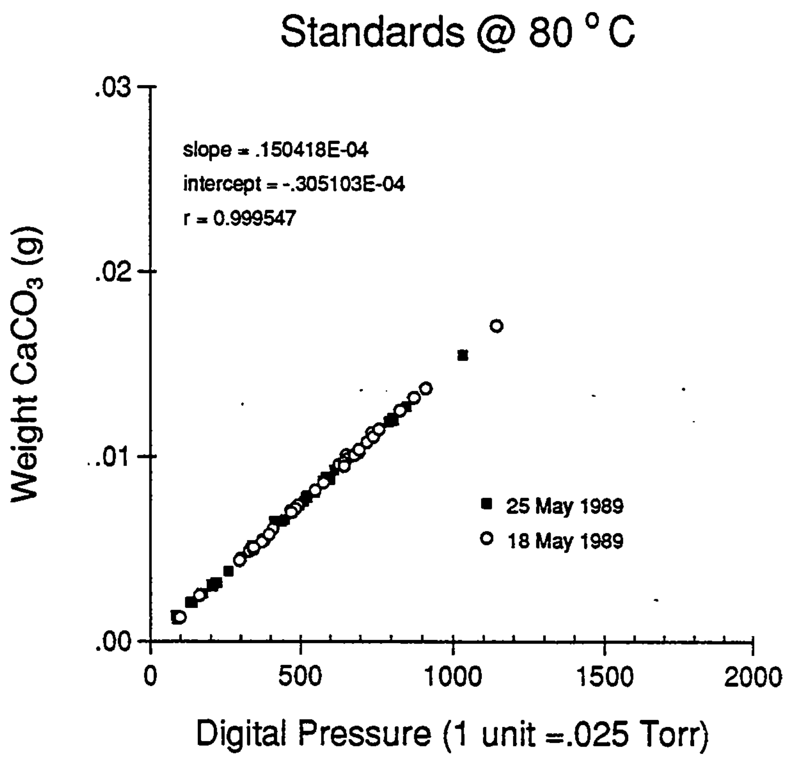


Figure 2: Two runs of 100% reagent grade calcium carbonate standards. The slope and intercept of the regression line produced from standards are used to calculate percent carbonate in unknown samples. For each new batch of phosphoric acid used, a new regression line is produced from standard runs.

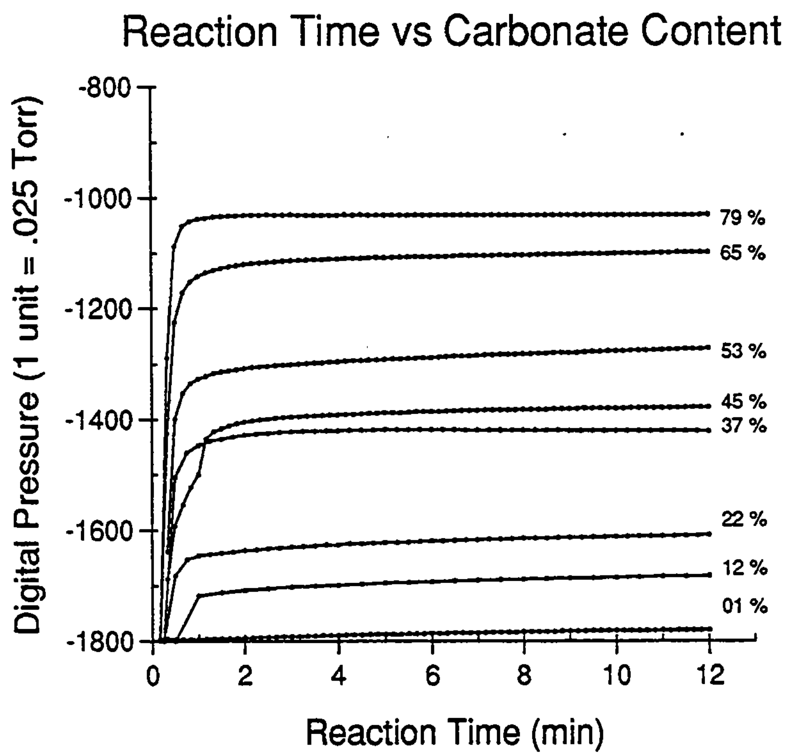


Figure 3: Reaction time for samples of varying carbonate content. For all samples, the bulk react of sample to CO_2 is complete within one minute. However, samples with less than 37% carbonte continue to react even after 12 minutes. The sediments are from cores recovered from the Sierra Leone Rise in the eastern equatorial Atlantic (Curry and Lohmann, 1985).

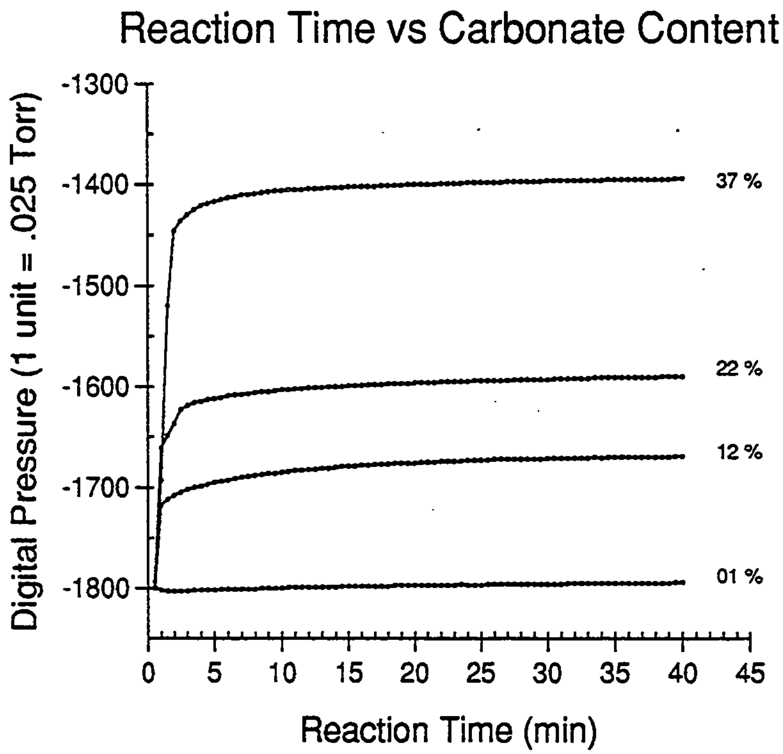


Figure 4: Reaction time for samples of low carbonate content. These samples were allowed to react for 40 minutes to find the reactime needed to evolve all the CO_2 from low carbonate samples. At 20 minutes, the reaction is complete and is 98-100% of the CO_2 value after 40 minutes. We believe that the small but gradual increase in carbonate content after 20 minutes is due to water vapor accumulation from acid outgassing. The sediments are from cores recovered from the Sierra Leone Rise in the eastern equatorial Atlantic (Curry and Lohmann, 1985).

e

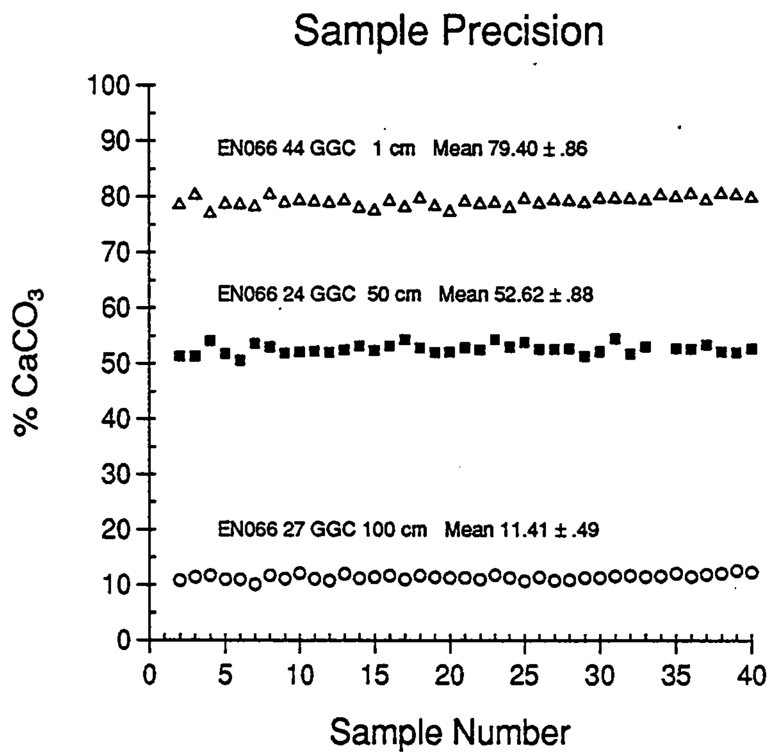


Figure 5: Sample precision versus carbonate content. We show 120 replicate analysis of carbonate for samples with high, medium and low concentrations. The sediments are from cores recovered from the Sierra Leone Rise in the eastern equatorial Atlantic (Curry and Lohmann, 1985).

Appendix 1. Board jumpering specifics.

Board Jumpering - Various options are available on each of the following boards and are selected using removable jumper plugs. Features are selected by installing or removing the jumper plugs as noted below. See the appropriate manuals for further explanation. Only jumpers which are to be installed are detailed unless specifically noted.

VL-7806 (Z80 CPU)

J4	Memory map select/segment signal connector
M0	thru M3 use 8K Eprom chips
V1	a - connects CTC output 1 to serial channel B baud input
V1	c - connects SYSCLK/2 to CTC input 1
V2	DCE to terminal
V3	DCE to terminal
V4	DCE to terminal
V5	DTE to computer
V6	DCE to terminal
V7	DTE to computer
V8	DTE to computer
V9	DTE to computer
V10	b & c - sockets M0-M3 are 8K RAM chips
V11	a, b, c, & d - sockets M0-M3 are all enabled
V12	b - MEMEX is set low at power up
V13	MEMEX signal is controlled on the board
V14	a - IOEXP is connected to ground
V14	b - AUXGND is connected to digital ground

DSC-7911 (Dual stepper controller)

J1	no jumper - no interrupt desired
J2	center pin to NE, non-expanded I/O map
J7	E1 tied to 5, clock frequency of 6.25 kHz
SW1	1-5 on, to be address 15 matching A3-A7
SW1	6 off, enables the card

Appendix 1. Board jumpering specifics continued.

VL-7709 (64/256K memory board)

A	M0 & M1 are 8K chips and are disabled
B	M2 & M3 are 8K chips and are disabled
C	M4 & M5 are 8K chips and M4 is enabled, M5 is disabled
D	M6 & M7 are 8K chips and are disabled
V2	a-f all installed for no bank switching
V3	c, e, & g - 8K chips used for a total of 64K
V4	d & h - bank control port address EE
V5	b - IOEXP is active low
V6	no jumpers installed - 16 bit addressing without bank control

VL-STD AIN-1 (integrating analog input board)

7	= address of the board
d	= differential input mode selected

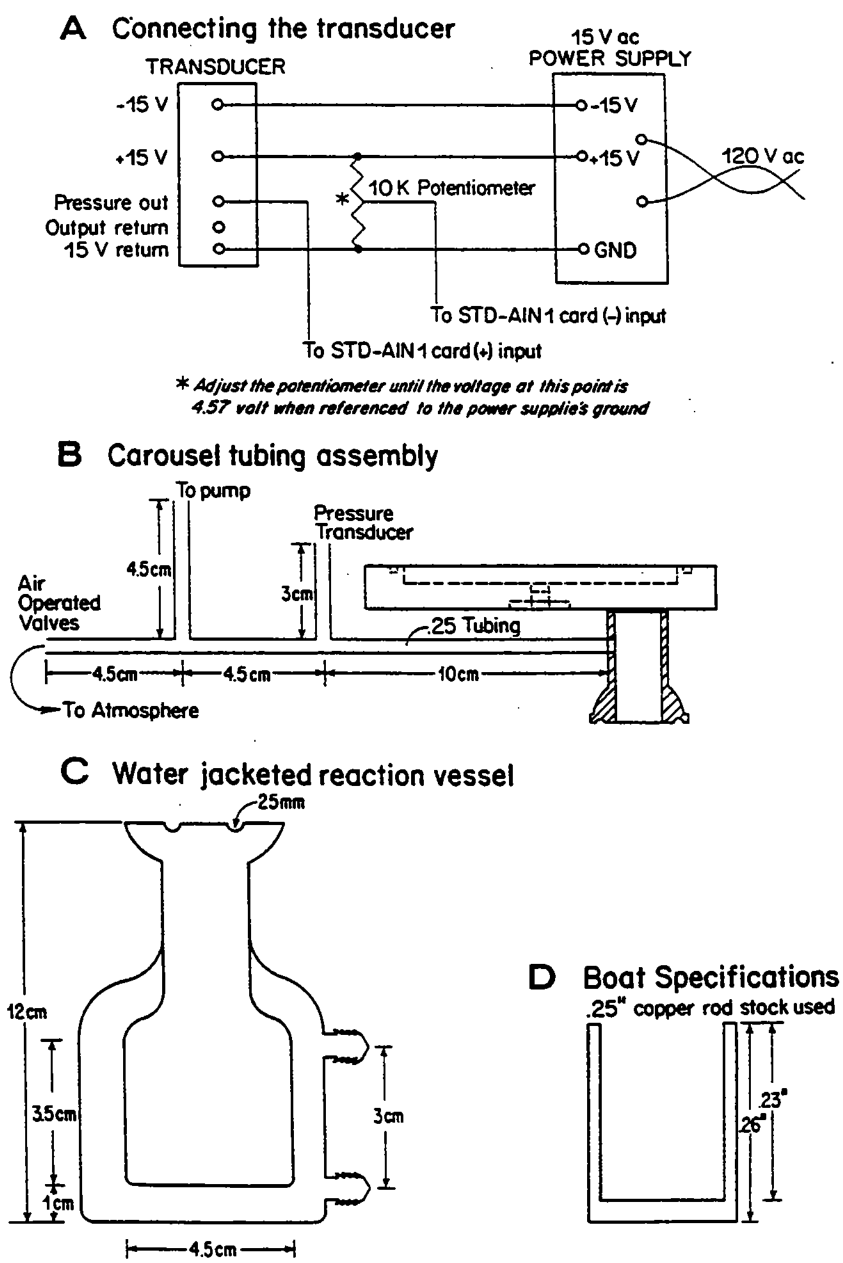
USD-7911 (Unipolar stepper driver)

E1	voltage range 8-32 volts
E2	external 5 volt supply not allowed
H3	Motor connections used
H3	pin 3 phase 4 output
H3	pin 4 phase 3 output
H3	pin 5 phase 2 output
H3	pin 6 phase 1 output
H3	pin 9 phase 1 common
H3	pin 11 motor supply voltage ground
H3	pin 12 motor supply voltage (+)

Appendix 2. Automated carbonate system memory map.

ADDRESS	K BYTES	NOTES
0000	0———————(	
	C4 BASIC ROM	
1000	4	Memory socket 0 on the Z80 board
2000	8———————(	
	System RAM	
3000	12	Memory socket 1 on the Z80 board
4000	16 (	
5000	20	Memory socket 2 on the Z80 board
6000	24 (	
7000	28	Memory socket 3 on the Z80 board
7400	29———————	
	Main Storage Area	
8000	32 (	
9000	36	Memory socket 4 on the memory board
A000	40 (	
B000	44	Memory socket 5 on the memory board
C000	48 (	
D000	52	Memory socket 6 on the memory board
DC00	55———————	
	Reserved for Machine Language Subroutines	
E000	56———————(	
	Not Used	
F000	60	Corresponds to socket 7 on the memory board (which is empty and disabled)
FFFF	64———————(	
	Top of Memory	

Appendix 3. System component diagrams.



Appendix 4. Phosphoric acid mixing procedure.

This procedure is modified from Coplen et al, 1983 and is the same procedure used in the WHOI mass spectrometer facility. The 200°C hot acid will remove the ink from the thermometer so the thermometer should be inserted into a glass tube sealed at one end before immersing into the acid. Once the acid has cooled, it can be stored in a glass bottle with a cap containing a conical insert for tight sealing.

Chemicals needed:

P_2O_5	Phosphorus pentoxide (2.01 kg)
H_3PO_4	Phosphoric acid (4.30 kg or 2.5 liters 85%)
CrO_3	Chromium (ic) oxide (10 mg)
H_2O_2	Hydrogen peroxide (3 ml 50%)

Procedure:

1. Place a 4 liter beaker on a hot plate in a fume hood, wrapping the beaker in aluminum foil. The foil will help the acid maintain a constant temperature while the fume hood is operating.
2. Mix together the phosphorus pentoxide, the phosphoric acid and the chromium oxide. The solution should be yellow.
3. Heat uncovered at 200°C for 7 hours.
4. Add the hydrogen peroxide.
5. Heat uncovered at 220°C for 4.5 hours. The original 3.5 liters will evaporate to 3.25 liters.
6. The specific gravity of the final solution should be between 1.90–1.92. If it is not, more phosphorus pentoxide should be added as required and the entire process is repeated from step 3.

Appendix 5. Complete program listings for the following:

Percent.for
Getdata.bas
Clear.bas
Animat.bas
Loader.bas

```
C
C     Percent.for
C
C     This program calculates the percent carbonate for samples
C      run on the Automated Carbonate System using a regression
C      line produced from reaction of 100% calcium carbonate
C      standards.
C
000    REAL SLOPE,INCPT
000    CHARACTER*64 FILEN,FILEM
000    CHARACTER Q
000    DATA SLOPE /.150418E-4/, INCPT /-.305103E-4/
000    WRITE(*,101) ' Input file: '
101    FORMAT(A,/)
000    READ(*,10) FILEN
010    FORMAT(A)
000    OPEN(UNIT=10,FILE=FILEN,STATUS='OLD')
031    WRITE(*,111) ' Slope: ',SLOPE
111    FORMAT(A,G15.6,/)
000    WRITE(*,101) ' Change? (y/N): '
000    READ(*,10) Q
000    IF(Q.eq.'Y'.or.q.eq.'y') THEN
000    WRITE(*,101) ' Enter new slope: '
000    READ(*,*) SLOPE
000    GO TO 31
000    ENDIF
032    WRITE(*,111) ' Intercept: ',INCPT
000    WRITE(*,101) ' Change? (y/N): '
000    READ(*,10) Q
000    IF(Q.eq.'Y'.or.q.eq.'y') THEN
000    WRITE(*,101) ' Enter new intercept: '
000    READ(*,*) INCPT
000    GO TO 32
000    ENDIF
000    WRITE(*,101) ' Output file: '
000    READ(*,10) FILEM
000    OPEN(UNIT=11,FILE=FILEM,STATUS='NEW')
001    READ(10,11,END=2)XNUM,DEPTH,WEIGHT,XPRESS
011    FORMAT(4F10.0)
C      IF(WEIGHT.LT..01)GO TO 1
000    TMP=XPRESS*SLOPE+INCPT
000    XPERC=TMP/WEIGHT*100.
000    WRITE(11,12)XNUM,DEPTH,XPERC
000    GO TO 1
002    CLOSE(10)
000    CLOSE(11)
012    FORMAT(F10.0,F10.2,F10.1)
000    STOP
000    END
```

```
C
C       Getdata.bas
C
C       This program produces paired data of pressure vs. time.
C
1       rem Getdata.bas produces pressure/time data
2       rem The boat must already be loaded in the slot to the
3       rem right of the delivery slot and the carousel must be
4       rem closed before running the program.
5       gosub 9549:rem close valves/pumpdown subroutine
6       rem (CAPS LOCK) must be depressed!
7       gosub 9525:rem close valves
10      OUT 01,50
20      OUT 00,250 : REM STARTING PULSE RARE
30      OUT 00,50 : REM HIGH SPEED PULSE RATE
40      OUT 00,&0F
50      out 00,&0F :rem larger deacceleration
100     print:print" enter # of samples ";
110     input l
120     print" enter sample interval (in secs.) ";
130     input q
140     OUT 01,77:OUT 00,254
500     gosub 1000
550     OUT 01,64
590     gosub 9500:rem open to atmosphere when done
600     print " Hit <P> to print the data or <Q> to quit =>";
610     A=USR(-9215)
620     if A=81 end
630     IF A=80 GOTO 3000
635     PRINT
640     GOTO 610
1000    REM * create pressure/time data
1010    for z=1 to l
1020    @(z)=ain(&70,0)
1030    gosub 2000
1040    next z
1050    return
1060    rem
2000    rem *delay
2010    delay Q*1000
2020    return
3000    rem * Print data
3010    for j=1 to l
3020    Z=J*q:GOSUB 6000:CPRINT 32:GOSUB 6300:GOSUB 4000:Z=@(J):GOSUB 6000
3021    GOSUB 6300
3025    CPRINT 13
3030    next j
3035    A=Usr(-9215)
3040    GOTO 600
4000    rem *do spacing
4010    for i=1 to 3
4020    CPRINT 32
4030    next i
4040    return
6000    REM DECIMAL ECT
```

Getdata.bas continued.

```
6005    REM This routine takes the var. Z and creats a
6010    REM 5 byte string of #'s ,containing the the ASCII codes for the
6015    REM characters which make up the printed representation of the # Z
6020    REM and stores them in D(0)-D(4)
6050    FOR X=0 TO 5:D(X)=32:NEXT X
6055    D(1)=48
6060    C=2:P=10000
6065    REM if Z is negitive then D(0)=45 (45=the ASCII code for "-")
6070    IF ABS(Z)<>Z D(0)=45:Z=ABS(Z)
6075    IF Z>9999 D(1)=49:Z=Z-10000
6100    D(C)=MOD(Z,P)/(P/10)+&30
6110    IF P=10 GOTO 6200
6120    P=P/10
6130    C=C+1
6140    GOTO 6100
6190    REM remove leading 0's and replace them with spaces
6200    FOR X=1 TO 4
6210    IF D(X)<>48 RETURN
6215    REM move sign 1 location to the right
6220    D(X)=D(X-1):D(X-1)=32
6230    NEXT X
6240    RETURN
6300    REM output contents of D(0)-D(4) to dart channel A
6305    D(6)=46:D(7)=48
6310    FOR X=0 TO 7
6330    CPRINT D(X)
6340    NEXT X
6350    RETURN
9549    print:print "Pumping on the acid for 27 minutes";
9500    rout 96,7,1:rem open to atmosphere at end
9501    return
9525    rout 96,6,0:delay 1000:rem close valve to pump
9526    return
9550    rout 96,7,0:rout 96,6,1:delay 9000:delay 9000:delay 9000:
9551    rem 9550 close valves/pumpdown on acid for 27 seconds
```

```
C
C    Clear.bas
C
C    This program clears the existing memory configuration so that
C    the storage area can be reinitialized.
C
05   rem Clear.bas, last revised 26.8.89 (DRO)
10   rem this program nukes the existing storage area so that NOVOS can be
20   rem re-initialized
30   rem CAUTION: this program destroys everything in the existing storage area
40   rem NOTE: after you run this the computer will lock up. Type '0' over and
50   rem over until it comes back to life (see page 2-6 of the C4-BASIC manual)
70   print " Enter the starting address of the storage area in hex ";
75   rem As presently configured, input '&7400'
80   input h
85   print " KABOOM !!!!!!!! HA HA HA HA !!!!!"
90   for z=0 to 50:poke h+z,0:next z
```

```
C
C      Animat.bas
C
C      This program runs the Automated Carbonate System.
C
C
1      rem The Animating Element, last revised 30.8.89 (DRO)
5      print:print" Welcome stranger.":print
10     rem hardware initialization
20     gosub 6800:rem initialize serial port B to talk to the Mettler
30     gosub 7300:rem initialize motor controller card
40     print" Skip software initialization? (Y/N)";
45     gosub 700
46     rem if "Y" then skip it
47     if a=89 goto 91
48     print:print " Initializing......"
50     rem init. variables
60     for z=0 to 816:@(z)=32:next z:rem clear out array
65     rem put zeros in the pressure value slots
70     for w=1 to 40:v2=0:gosub 9310:next w
80     N=1:rem sample # of the next sample 2b weighed
81     L=40:rem # of boats
83     for z=0 to 7:f(z)=0:next z:rem zero all the flags
85     l2=5:rem minimum pressure value (used in run routine)
91     print:print" Is the Lazy Susan homed to slot #1 ? (Y/N)";
92     rem call up Homer menu if "Y"
93     gosub 700:if a=78 gosub 5100:goto 100
95     c=1
100    print:print" The Animating Element"
110    print:print
120    print" Type :"
130    print" <W> to weigh samples"
132    print" <L> to load the sample boats"
135    print" <R> to run samples"
150    print" <P> to print the data"
155    print" <I> to re-initialize"
156    print" <T> to tweak the system"
160    print" <Q> to quit"
170    print:print:print
180    print" =>";
190    gosub 700:rem call a ML single char input sub.(A=the ASCII val of char)
195    rem ASCII codes:87="A",76="L",82="R",80="P",73="I",84="T",81="Q"
199    rem weigh samples
200    if a=87 gosub 1000:goto 110
219    rem load boats
220    if a=76 gosub 2000:goto 110
224    rem run samples
225    if a=82 gosub 3000:goto 110
229    rem print data
230    if a=80 gosub 4000:goto 110
234    rem re-initialize
235    if a=73 goto 5000
240    rem call up system tweak menu
245    if a=84 gosub 500:goto 110
248    rem quit
249    if a=81 goto 300
250    goto 190
```

```
300    rem quit
310    print:print" Quitting will destroy all the data in memory. Are you sure"
320    print"you want to quit (Y/N) ?";
330    gosub 700:if a<>89 goto 100
340    print:print" goodbye.....":end
490    rem Tweak the system menu and options
500    print:print" Tweak the system........":print
510    print" Hit :"
520    print" <H> to home the Lazy Susan"
530    print" <A> for atmosphere"
540    print" <V> for vacuum"
545    print" <W> to re-weigh an entry"
550    print" <R> to continue running after a jam up"
570    print" <C> to check the pressure in the vessel"
580    print" <E> to exit to the main menu"
585    print:print:print
590    print" =>";
600    gosub 700:rem get a single keypress
605    rem ASCII codes:72="H",65="A",86="V",87="W",82="R",67="C",69="E"
609    rem call homer subroutine
610    if a=72 gosub 5100:goto 500
615    rem open to atmosphere
620    if a=65 goto 9500
629    rem pump down
630    if a=86 goto 9550
632    rem re-weigh
634    if a=87 gosub 1500:goto 500
639    rem continue running
640    if a=82 gosub 3300:goto 110
659    rem check pressure
660    if a=67 gosub 8400:goto 500
669    rem return to main menu
670    if a=69 return
680    goto 600
700    rem*******Get a single Keypress, Convert lowercase letters to Caps.*****
710    a=usr(-9215)
720    if a<97 return
730    if a>122 return
740    a=a and &DF: rem convert lowercase => uppercase
750    return
1000   rem******************* weigh samples ************************************
1002   rem This sub. prompts the user to place the boats and the samples
1005   rem on the balance,tares the boats, weighs the samples and stores
1010   rem the results in array @(x)
1020   if f4=1 print" All the samples have already been weighed.":return
1025   if f6=1 print" Sorry you can not resume weighing after loading.":return
1100   print:print
1101   print" "
1102   print:print "Sample # ";n
1105   print
1106   print" Enter the depth in cm => ";
1107   rem get depth value
1108   w=n:gosub 9010:if f1=1 f1=0:return
1109   print
1110   print" Hit <spacebar> when you are ready to tare the boat.=>";
1115   rem call ML character input sub.
1117   rem if inputted character = <CTRL> Z return
```

```
1120 gosub 700:if A=26 return
1121 if a=4 goto 1105
1122 if a<>&20 goto 1120
1123 rem 4='D̂',&20='<SPACEBAR>'
1125 rem
1127 print:print "taring...."
1128 rem tare, exit if escape flag set
1130 gosub 6600:if f1=1 f1=0:return
1135 rem
1140 print:print"Tare completed."
1150 print:print" Add the sample to the boat and hit any key when it is"
1160 print" ready to be weighed.=>";
1165 rem wait for a keypress. return if <CTRL><Z>
1170 gosub 700:if a=26 return
1171 rem if there was a <CTRL><T> retare.
1172 if a=20 goto 1127
1173 rem
1175 print:print"Weighing"
1179 rem ask balance for next stable weight
1180 gosub 6700:rem S command
1185 rem
1186 rem input data from balance:if no errors occured goto 1200
1190 gosub 6010:if S=0 goto 1200
1192 print" An input error has occured.Type <R> for retry or <A> for"
1193 print" abort.=>";
1194 rem get keypress:return if <CTRL><Z>
1195 gosub 700:if A=26 return
1196 rem return if "A"
1197 if a=65 return
1198 goto 1180
1200 w=n:gosub 9210: rem store weight
1210 w=n:gosub 9410: rem print data
1222 print
1223 print" Do you wish to redo this sample ? (Y/N)";
1224 rem get keypress: return if <CTRL><Z>
1225 gosub 700:if A=26 return
1226 rem if "Y" redo entire sample,if "Ŵ" re-weigh
1227 if A=89 goto 1100
1228 if A=23 goto 1175
1229 rem return to re-weigh function if re-weigh flag set
1230 if f0=1 return
1232 rem set flag and return if all samples weighed
1234 if n=l f4=1:return
1240 n=n+1
1300 goto 1100
1500 rem ****************** re-weigh a sample *******************************
1505 if n=1 print " No samples have been weighed yet.":return
1507 t=n:rem save old value
1510 print:print:print" Enter the number of the sample to re-weigh (0 aborts)";
1520 input n
1530 if n=0 goto 1570
1540 if n<1 goto 1510
1550 if n>t print" That sample hasn't been weighed yet.":goto 1510
1555 rem set "re-weigh" flag, call weigh subroutine
1560 f0=1:gosub 1100:f0=0
1565 rem restore old value
1570 n=t:return
```

```
2000 rem ***************** load boat onto lazy-susan ************************
2005 if f4=0 print:print" All the samples have not been weighed yet":goto 100
2007 d2=50:rem motor speed, used in routine at 7200
2009 f6=1:rem flag set when load routine called
2010 goto 9500
2015 if f5=1 print:print" Boats already run.....":return
2020 if f3=1 print:print" boats already loaded....":return
2040 l=n
2050 print
2055 rem keep from dumping 1st boat when loading 40th
2060 if f2=1 c=40
2070 print" Load the boat into load slot";c;" and hit the <SPACEBAR> =>";
2080 if f2=1 c=39
2085 rem get keypress: return if <CTRL><Z>
2090 gosub 700:if a=26 return
2100 if a<>32 goto 2090
2105 rem load 40th boat if f2 set and set f3 (all boats loaded flag)
2110 if f2=1 print:print" Boat ";40;" loaded.":f3=1:return
2120 if c=39 f2=1:c=40:goto 2140
2130 d=0:t=c+1:gosub 7010:c=t:rem go to hole T subroutine
2140 print:print" Boat ";c-1;" loaded."
2145 if f2=1 c=39:rem more flag gymnastics
2147 rem keep loading if there are more boats
2150 if c<l goto 2050
2170 return
3000 rem********************** run samples ********************************
3005 rem choose the reaction time in minutes
3010 gosub 9600
3020 if f3=0 print" The boats have not all been loaded yet.":return
3034 if f1=1 f1=0:goto 3060
3035 if f5=1 print" The samples have already been run.......":return
3036 f5=1
3040 if c<>39 t=39:d=0:D2=20:print " advancing to home position...":gosub 7010
3045 D2=200:rem motor speed delay used in routine at line 7200
3050 c=39
3060 l=n
3065 if l<40 l=l-1
3070 print" pumping down";
3075 gosub 8010: rem call pumpdown subroutine
3080 rem get pressure, return if escape flag set.
3085 gosub 8100: if f1=1 return
3090 v1=a: rem store average baseline pressure value
3100 print" done."
3110 r=c+2:if r>40 r=r-40
3120 t=c+1:d=0:gosub 7010
3130 print:print" running sample #";r;
3135 rem minutes changed to seconds reaction time
3140 k=K1*60
3145 rem read pressure, return if escape flag set
3150 gosub 8100:if f1=1 return
3160 v2=a-v1: rem get change in pressure
3170 w=r:gosub 9310: rem save pressure value in array
3180 print" done.":print
3190 w=r:gosub 9410:rem output pressure data to terminal
3200 c=c+1:if c>40 c=c-40
3202 if v2>l2 goto 3210
3204 for z=0 to 20:cprint(7):next z
```

```
3206	print:print" Check to make sure the Lazy-susan isn't jammed."
3207	print" Hit <A> to abort or <R> to resume =>";
3208	gosub 700:if a=65 f1=1:return
3209	if a<>82 goto 3208
3210	if r< 1 goto 3070
3220	if c<>1 print" advancing to load slot #1.....":gosub 7100
3230	c=1
3235	rem at end of run put system under vacuum
3236	goto 9550
3240	return
3300	rem********** continue running after a jam up **************************
3310	print:print
3312	t=c: rem save old value in case user aborts
3315	print " Enter the # of the load slot the L.S is homed to (0 to abort) ";
3320	input c
3325	if c=0 c=t:return
3326	if c>40$  to 3320
3327	if c<1 goto 3320
3330	f1=1
3340	gosub 3000
3350	return
4000	rem*******print contents of weight/pressure file***********
4010	rem
4020	if n=1 print:print" Sorry, there is nothing to print.":return
4030	if n<40 n=n-1
4050	for w=1 to N
4060	gosub 9410
4070	next w
4080	if N<40 N=N+1
4090	gosub 700:rem wait for a key press
4100	return
5000	rem ******************* reset sample pointer ***************************
5010	print:print" CAUTION: re-initializing will destroy all the"
5020	print" data in memory. Are you sure you want to do it (Y/N) ?";
5030	gosub 700:if a<>&59 goto 110
5040	print:print" Forgetting.........":goto 50
5100	rem******************* home the lazy susan ******************************
5105	if f3=1 print:print:print" CAUTION: boats have already been loaded!"
5110	print:print:print" Home Menu"
5120	print:print
5140	print" < RIGHT ARROW > = counter clockwise"
5250	print" < LEFT ARROW > = clockwise"
5260	print" < DOWN ARROW > = single step mode"
5270	print" < UP ARROW > = continuous step mode"
5280	print" < S > = Smart Home option"
5290	print" < E > = exit"
5300	print:print " =>";
5310	gosub 700: rem get a single keypress
5320	rem f7= mode flag, 0 = single step mode, 1 = continuous step mode
5330	out 01,64:rem send stop command to motor controller card
5340	if a=10 f7=0: rem when booted under Async
5350	if a=24 f7=0: rem when booted under Word Star
5360	if a=11 f7=1: rem Async
5370	if a=5 f7=1: rem WS
5380	if a=29 d=1:gosub 5500:rem Async
5390	if a=19 d=1:gosub 5500:rem WS
5400	if a=12 d=0:gosub 5500:rem Async
```

```
5410    if a=4 d=0:gosub 5500:rem WS
5415    rem smart home option
5420    if a=83 goto 5600
5425    rem return to other menu
5430    if a=69 goto 5700
5440    goto 5310
5500    rem send commands to motor controller card
5510    if f7=1 goto 5550
5520    out 01,66 or (d*8): rem single step,direction d
5530    return
5550    out 01,69 or (D*8): rem continuous step mode, direction d
5555    out 00,254: rem motor speed
5560    return
5570    rem* smart home option
5600    print:print" Enter the current load slot # ";:input c
5602    if c<1 goto 5600
5604    if c>40 goto 5600
5610    gosub 7100:return
5700    rem* return to main menu
5710    print:print" Enter the current load slot # ";:input c
5712    if c<1 goto 5710
5714    if c>40 goto 5710
5720    return
6000    rem**********Mettler I/O subroutines************
6010    rem Generic input subroutine
6020    rem This routine calls a m/l routine to input data from the Mettler
6030    rem and sets the flag S on the result.
6040    rem if S=0 the data was succesfully inputed and is stored in
6050    rem @(0)-@(15)
6060    rem if S=1 a tare occurred and the data in @(0)-@(15) is invalid
6070    rem if S=2 no tare occured and no data was received
6080    S=0
6090    rem clear out input subroutines buffer
6100    for Z=0 to 15
6110    POKE -9201+27+Z,32
6120    next Z
6130    rem call machine language Mettler I/O subroutine
6140    A=USR(-9201)
6150    rem take data from input subroutine's buffer & put it in @(0)-@(15)
6160    for Z=0 to 15
6170    @(Z)=PEEK(-9201+27+Z)
6180    next Z
6190    rem check to see if first data byte valid (&53="S")
6200    if @(0)=&53 goto 6260
6220    rem first byte isnt the start of a data block:goto interpret error sub
6230    goto 6280
6240    rem check to see if last byte valid (&A=<lf>).if so assume
6250    rem the whole data block is valid
6260    if @(15)=&A return
6270    rem find location of <lf> in array @(X)
6280    for Z=0 to 15
6290    if @(Z)=&A goto 6340
6300    next Z
6310    rem if no <lf> found set error flag and return
6320    S=2:return
6330    rem check first char of message CASE "T"->6400
```

```
6340    if Z<3 S=2:return
6350    if @(Z-3)=&54 goto 6400
6360    rem if message doesn't start with any of above chars it is invalid
6370    rem set error flag and return
6380    S=2 :return
6390    rem if second char of message = "A" set error flag=1 (1=tare)& return
6400    if @(Z-2)=&41 S=1:return
6410    rem if first char of message ="T" & 2nd <> "A" then the message is
6420    rem invalid,set error flag and return
6430    S=2:return
6490    rem--------------------------------------------
6500    rem disable tare bar ie. send R1<cr><lf> to the balance
6510    gosub 6900:out &F6,&52
6520    gosub 6900:out &F6,&31
6530    gosub 6900:out &F6,&0D
6539    gosub 6900:out &F6,&0A
6550    return
6555    rem--------------------------------------------
6560    rem enable tare bar ie. send R0<cr><lf> to the Mettler
6570    gosub 6900:out &F6,&52
6580    gosub 6900:out &F6,&30
6590    gosub 6900:out &F6,&0D
6593    gosub 6900:out &F6,&0A
6597    return
6598    rem--------------------------------------------
6600    rem send tare command ie. send T<cr><lf> to the Mettler
6605    Q=0:f1=0
6610    gosub 6900:out &F6,&54
6615    gosub 6900:out &F6,&0D
6620    gosub 6900:out &F6,&0A
6625    gosub 6010:rem error check routine
6630    if S=1 return
6635    rem if the balance doesn't tare correctlly retry up to 4 times
6640    Q=Q+1:if Q<4 goto 6610
6645    print:print
6650    rem if the balance still hasn't tared give the user a chance to abort
6655    print" The balance refuses to tare.Type <R> for retry or <A> for
6660    print" abort.=>";
6665    rem wait for a keypress:if the char ="R" then retry
6670    gosub 700:if A=&52 goto 6605
6675    rem ifthe char="A" then return to the menu
6680    if A=&41 f1=1:return:rem set escape flag and return
6685    goto 6670
6690    rem--------------------------------------------
6700    rem ask balance for next stable weight (ie. send S<cr><lf> to balance)
6710    gosub 6900:out &F6,&53
6720    gosub 6900:out &F6,&0D
6730    gosub 6900:out &F6,&0A
6740    return
6745    rem--------------------------------------------
6800    rem initialize serial port B
6805    out &F1,&45
6810    out &F1,&D0
6815    out &F7,&10
6820    out &F7,&18
6825    out &F7,&4
6830    out &F7,&47
```

```
6835  out &F7,&05
6840  out &F7,&AA
6845  out &F7,&03
6850  out &F7,&41
6855  out &F7,&00
6860  return
6870  rem--------------------------------------------------
6900  rem do output handshaking
6910  rem see p3-14 of the Z80 ref manual
6920  if RIN(&F7,2)=0 goto 6920
6930  if RIN(&F7,5)=0 goto 6930
6940  return
7000  rem*********** motor controller card subroutine ***********************
7010  rem * GO to HOLE T, IN DIRECTION D, FROM CURRENT HOLE, C
7020  if D=1 goto 7060
7030  N2=(T-C)*50
7040  if N2<0 N2=N2+2000
7050  goto 7080
7060  N2=(C-T)*50
7070  if N2<0 N2=N2+2000
7080  gosub 7200
7090  return
7095  rem--------------------------------------------------
7100  rem *home lazy-Susan to load slot #1
7110  if c>=20 d=0
7120  if c<20 d=1
7130  t=1:d2=50:gosub 7010
7140  c=1
7150  return
7190  rem--------------------------------------------------
7200  rem *GO N STEPS IN DIRECTION D
7210  for Z= 1 to N2
7220  out 01,66 OR (D*8)
7230  delay d2
7240  next Z
7250  return
7290  rem--------------------------------------------------
7300  rem initialize motor controller card
7310  out 01,50
7320  out 00,250
7330  out 00,50
7340  out 00,&0f
7350  out 00,&0f
7360  return
8000  rem******************* valve card and transducer routines **************
8020  goto 9550
8023  rem pump down to 4.2 Torr
8025  if ain(&70,0)>-1780 goto 8025
8040  goto 9525
8050  return
8090  rem--------------------------------------------------
8100  rem *wait for pressure to settle
8105  rem o=0
8110  rem for z=1 to 500
8120  rem x=ain(&70,0)
8130  rem if abs(x-o)>3 print".";:o=x:goto 9510
8140  rem next z
```

```
8200 rem * average data from pressure transducer
8201 rem by making sure 10 values in a row are equal
8202 rem with a delay of 6 seconds between each reading
8205 b1=ain(&70,0)
8207 if ain(&70,0)>16000 goto 8260
8210 b=0
8215 delay 6000
8220 b2=ain(&70,0)
8225 if abs(b2-b1)>6 goto 8205
8230 b=b+1
8235 if b<9 goto 8215
8240 a=b1
8250 return
8260 out 120,0:delay 3000: rem close all valves
8262 print" The reaction chamber is open to the atmosphere."
8263 print" <P>umpdown or <A>bort ?";
8265 gosub 700
8270 if a=65 f1=1:return
8275 if a=80 out 120,2:delay 3000:goto 8205
8280 goto 8265
8300 rem delay k seconds
8310 for m=1 to k
8320 delay 1000
8330 next m
8340 return
8400 rem*************Check pressure******************************************
8403 gosub 9550:rem close valves/pumpdown subroutine
8404 print "Pumping on the acid for 36 seconds";
8405 delay 9000:delay 9000:delay 9000:delay 9000
8407 gosub 9525:rem close all valves
8410 print:print" Enter time interval in seconds (0 aborts) ";
8420 input k:if k<0 goto 8410
8425 print:"Getting initial pressure (1 minute)...";
8440 if k=0 return
8450 gosub 8100: v1=a: rem get initial pressure
8460 print " Waiting ";k;" seconds.......";
8470 gosub 8300: rem wait k seconds
8475 print: "Getting final pressure (1 minute)....";
8480 gosub 8100: rem get pressure again
8490 print" The change in pressure was ";v1-a;" digital units."
8492 gosub 9550:rem continue pumping on acid
8495 return
8500 rem*************Decimal output routines and stuff***********************
8505 rem This routine takes the var. Z and creats a
8510 rem 5 byte string of #'s ,containing the the ASCII codes for the
8515 rem characters which make up the printed representation of the # Z
8520 rem and stores them in J(0)-J(4)
8550 for X=0 to 4:J(X)=32:next X
8560 Q=1:P=10000
8565 rem if Z is negative then J(0)=45 (45=the ASCII code for "-")
8570 if ABS(Z)<>Z J(0)=45:Z=ABS(Z)
8600 J(Q)=MOD(Z,P)/(P/10)+&30
8610 if P=10 goto 8700
8620 P=P/10
8630 Q=Q+1
8640 goto 8600
8690 rem remove leading 0's and replace wAth spaces
```

```
8700 for X=1 to 3
8710 IF j(X) <>48 RETURN
8715 rem move sig@ 1 location to the right
8720 J(X)=J(X-1):J(X-1)=32
8730 next X
8740 return
8800 rem output contents of J(0)-J(4) to dart channel A
8810 for X=0 to 4
8830 Cprint J(X)
8840 next X
8850 return
9010 rem get a 10 byte striAg and save it in array @() at address w
9015 for z=0 to 9:@((w*20)+z)=32:next z
9020 for z=0 to 10
9025 f1=0: rem escape flag
9030 rem get a single character using ML routine at -9215 (dec)
9040 gosub 700
9045 if a=26 f1=1:return
9050 rem if it was a backspace then backspace
9055 if z=0 goto 9075
9060 if a=8 z=z-1:@((w*20)+z)=32:cprint 8:cprint 32:cprint 8:goto 9040
9070 rem store data in array
9075 if a=8 goto 9040
9076 rem if that character was a CR exit routine
9077 if a=13 goto 9160
9080 @((w*20)+z)=a
9110 rem wait for a CR or backspace after 10th char.
9120 if z=10 goto 9040
9130 rem print character
9140 cprint a
9150 next z
9160 if z=10 return
9162 if z=0 @((w*20)+9)=48:return
9165 for j=z to 0 step -1
9170 @((w*20)+j+10-z)=@((w*20)+J)
9180 @((w*20)+j)=32
9190 next j
9195 return
9210 rem store weight
9220 for z=0 to 8
9230 @((w*20)+z+10)=@(z+3)
9235 next z
9240 return
9310 rem store pressure
9320 @((w*20)+19)=v2
9330 return
9410 rem print a single entry
9412 cprint 32:cprint 32:cprint 32
9415 z=w:gosub 8500:gosub 8800:cprint 46:cprint 48
9420 for y=0 to 18
9430 cprint @((w*20)+y)
9440 if y=9 cprint 32
9450 next y
9460 cprint 32:cprint 32:cprint 32
9470 z=@((w*20)+19):gosub 8500:gosub 8800:cprint 46:cprint 48
9480 cprint 13
9490 return
```

```
9500 rem ****************open to atmosphere subroutine*****************
9505 rout 112,2,0
9510 rout 112,3,0
9515 delay 1000:rout 112,2,1
9520 return
9525 rem **********************close all valves subroutine***************
9530 rout 112,2,0
9535 rout 112,3,0
9540 delay 3000
9545 return
9550 rem ***********close valves/pumpdown subroutine********************
9555 rout 112,2,0
9560 rout 112,3,0
9565 delay 3000:rout 112,3,1
9570 return
9600 rem******************reaction time menu**********************
9610 print:print " Run the system............":print
9620 print " Choose the appropriate reaction time ":print
9630 print " Hit: "
9640 print " <L> low carbonate ( 0-25%), 20 minutes"
9660 print " <H> high carbonate (25-100%), 5 minutes"
9680 print:print:print
9690 print " => ";
9700 gosub 700: rem get a single keypress
9710 rem ASCII codes: 72='H',76='L',77='M'
9720 rem low carbonate
9730 if a=76 K1=20:goto 9800
9760 rem high carbonate or standards
9770 if a=72 K1=5:goto 9800
9800 return
9805 rem*********HARDWARE DEPENDANT SOFTWARE**********************
9810 rem
9820 rem There are a few hardware dependant lines in Animat.bas
9830 rem which will have to be changed if the hardware is
9840 rem configured differently. The lines are listed below and
9850 rem are marked with a '#' in the program listing.
9860 rem
9870 rem Relay output channels: 9505-9515, 9530-9535, 9555-9565
9880 rem Analog input address : 8025, 8120, 8205-8207, 8220
9890 rem Motor driver         : 5330, 5510-5560, 7310-7350
```

```
C
C     Loader.bas
C
C     This program loads two machine language subroutines into memory.
C
C
1     REM RDKEY & METIO loader program, last revised 29.8.89 (DRO)
2     rem This program loads 2 machine language subroutines,
3     rem RDKEY & METIO, into memory
4     rem RDKEY starts at -9215
5     rem METIO starts at -9201. It's buffer starts at -9174
6     x=&DC00
7     b1=&2A:b2=&dC
10    REM Single character input subroutine (RDKEY)
20    REM DB,F5 LOOP INA,F5H ;check RCV bit
30    REM CB,47 BIT 0,
40    REM 28,F9 JR Z,P ;i's not set wait
50    REM DB,F4 IN A,H ;g character
60    REM E6,7F AND 7F;lose the highest bit
70    REM 6F L t n L so basic can read it
80    REM 26,00 LD H,zero H so basic doesn't get garbage
90    REM C9 RTrn
100   REM poke single character input sub. into memory
110   X=X+1:POKE X,&0DB
120   X=X+1:POKE X,&0F5
130   X=X+1:POKE X,&0CB
140   X=X+1:POKE X,&047
150   X=X+1:POKE X,&028
160   X=X+1:POKE X,&0F9
170   X=X+1:POKE X,&0DB
180   X=X+1:POKE X,&0F4
190   X=X+1:POKE X,&0E6
200   X=X+1:POKE X,&07F
210   X=X+1:POKE X,&06F
220   X=X+1:POKE X,&026
230   X=X+1:POKE X,&000
240   X=X+1:POKE X,&0C9
250   REM Balance input subroutine (METIO)
260   REM b1 and b2 are variables
270   REM 06,10 START LD B,16 16=the maximum # of chars.
280   REM DD,21,b2,b1 LD IX,$b1b2 load IX with start of buffer
290   REM DB,F7 LP IN A,($F7) load A with status register
300   REM CB,47 BIT 0,A test bit 0 (bit=0 RCV flag)
310   REM 28,FA JR Z,LP if it=0 goto LP
320   REM DB,F6 IN A,($F6) load a with data register
330   REM E6,7F AND 7F
340   REM DD,77,00 LD (IX),A store A in buffer
350   REM DD,23 INC IX increment buffer pointer
360   REM FE,0A CP $A compare A to 10 (10=lf)
370   REM C8 RET Z if equal return to BASIC
380   REM 10,EC DJNZ LP decrement B and goto LPifnot0
390   REM C9 RET if b=0(ie.16 chars.have been
400   REM entered) return to BASIC
410   REM &F7= the status register of DART channel b
420   REM &F6= the data register of the same.
430   REM See pp 3-8 - 3-17 of the MULTIFUNCTION Z80CPU CARDS VL-7806 &VL-7807
440   REM for more info. on the dart chip. esp 3-14
```

Loader.bas continued.

```
460   rem poke balance input subroutine into memeory
470   X=X+1:POKE X,&06
480   X=X+1:POKE X,&10
490   X=X+1:POKE X,&DD
500   X=X+1:POKE X,&21
510   X=X+1:POKE X,b1: rem tell ML program where its buffer starts
520   X=X+1:POKE X,b2
530   X=X+1:POKE X,&DB
540   X=X+1:POKE X,&F7
550   X=X+1:POKE X,&CB
560   X=X+1:POKE X,&47
570   X=X+1:POKE X,&28
580   X=X+1:POKE X,&FA
590   X=X+1:POKE X,&DB
600   X=X+1:POKE X,&F6
610   X=X+1:POKE X,&E6
620   X=X+1:POKE X,&7F
630   X=X+1:POKE X,&DD
640   X=X+1:POKE X,&77
650   X=X+1:POKE X,&00
660   X=X+1:POKE X,&DD
670   X=X+1:POKE X,&23
680   X=X+1:POKE X,&FE
690   X=X+1:POKE X,&0A
700   X=X+1:POKE X,&C8
710   X=X+1:POKE X,&10
720   X=X+1:POKE X,&EC
730   X=X+1:POKE X,&C9
```

Appendix 6. System Configuration Procedure.

The system has a total of 48K RAM; three 8K RAM chips on the Z80 board and three 8K RAM chips on the memory expansion board. The present configuration is diagrammed in Figure 5 and is detailed as follows:

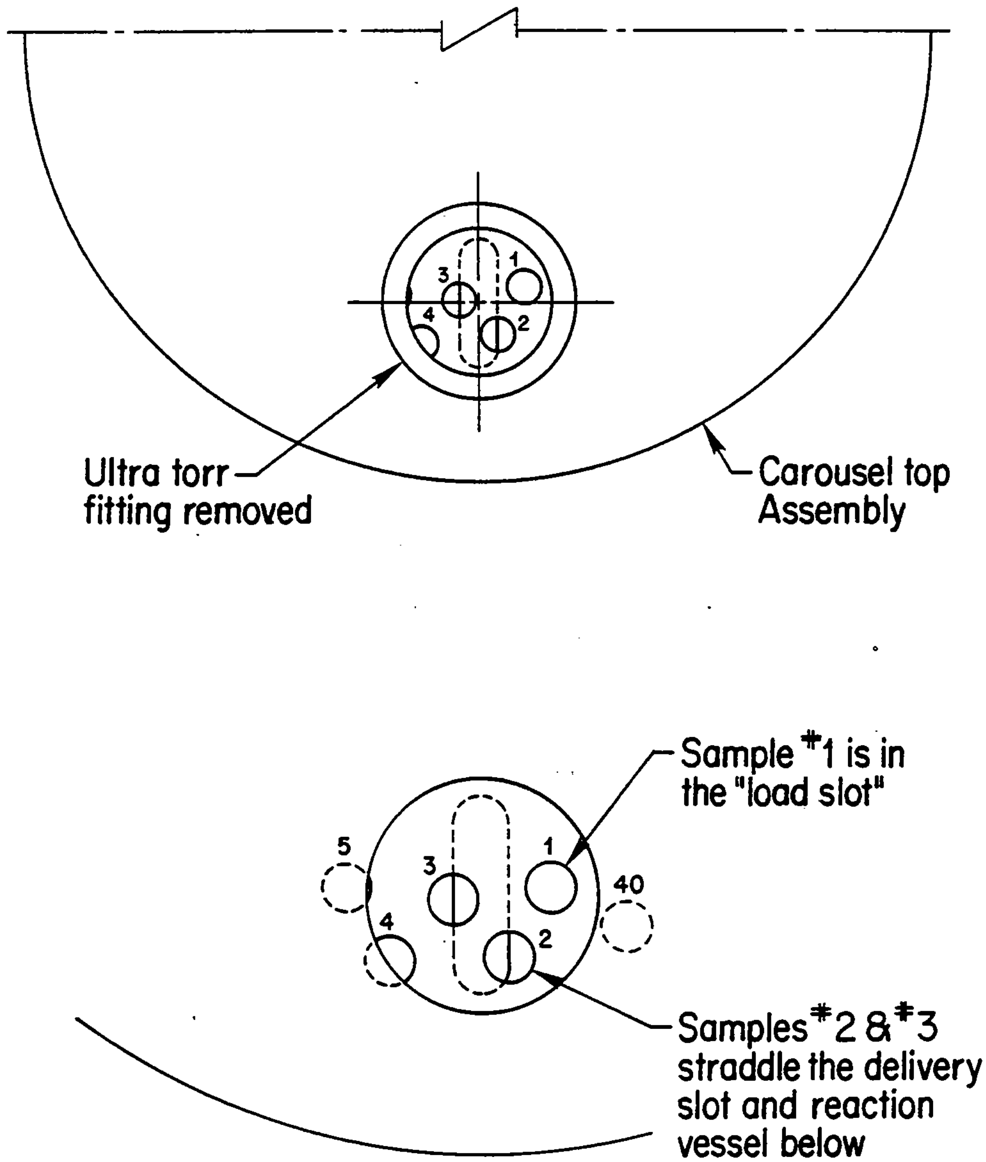
a) the storage area starts at &7400
b) the development RAM size is set to 21K
c) the storage area size is set to 26K
d) 1K at the top of memory is reserved for important machine language subroutines

If your system is not configured to reserve the necessary 1K of memory, reinitialize the system by following the procedure outlined below:

1. Enter *NOVOS* by typing '*NOVOS*'
2. Download *clear.bas* and run it (see *clear.bas* under **SOFTWARE**, p. 6)
3. Power down the system for 15 seconds and reset the power supply card
4. Turn the power back on and hit '0' (zero) repeatedly until the prompt appears
5. At the prompts enter;
   '21' for RAM size
   '0' for baud rate
6. Type '*NOVOS*'
7. Type '*SETUP*'
8. At the prompts enter;
   '1' for memory type (CMOS RAM)
   '7400' for the start of the storage area
   '26' for the storage area size
   '5' for the number of directory entries
9. Download *loader.bas* and run it (see *loader.bas* under **SOFTWARE** p 7.)
10. Download *animat.bas* and fix the downloading errors if any
11. Save *animat.bas* in slot 1 of the directory by typing 'SAVE 1'
12. Check to see that the machine language subroutines are properly loaded.
    Type 'PRINT USR(-9215)'<RET>
    Hit the space bar
    '32' should be printed on the screen (32 is the ASCII code for 'space bar')
    If not, run *loader.bas* and begin from step 9
13. Type 'DIR' to see that the following is printed:
    *DIR
    1 AUTOMATED CARBONATE SYSTEM program, last revised 16.5.89 (DRO)
    2 RDKEY & METIO LOADER program, last revised 26.8.89 (DRO)

RDKEY and METIO should now be resident in memory, starting at &DC01 (-9201) decimal. The execution addresses are -9215 for RDKEY and -9201 for METIO. METIO'S buffer, containing the character string inputed from the balance, begins at -9174 and is 16 bytes long. Whenever the system is powered up, always first run the loader program ('run 2') followed by the AUTOMATED CARBONATE SYSTEM program ('run 1') to assure that RDKEY and METIO are properly loaded.

Appendix 7. Sample carousel load slot diagram.



Appendix 8. Possible suppliers of equipment for the ACS

COMPANY	SPECIFIC SUPPLIES NECESSARY
Acculex 440 Myles Standish Blvd. Taunton, MA 02780	#DP-654 Voltage input meter
Ace Glass Inc. 1430 N.W. Blvd Vineland, NJ 08360-688	#9519-10 Pinch clamp for 25mm joint
All-Stainless 75 Research Rd. Hingham, MA 02043	10-32 x 3/8" allen cap screws 18-8 s.s. 10-32 x 2" allen cap screws 18-8 s.s
Analog Devices 1 Technology Way Norwood, MA 02062-9106	chassis mounted ac/dc power supply +/- 15 vdc output #952
Anderson Glass Old Turnpike Rd. R.F.D. 1 Fitzwilliam, NH 03447	glass reaction vessel; see Appendix 3.
B & D Motor Control Corp 99 Lowell Road Hudson, NH 03051	Slo-Syn DC stepping motor #MO61-FD-311
Cambridge Valve & Fitting 50 Manning Rd. Billerica, MA 01821	Cajon ultra-torr union fitting 1/2" to 1/4" #SS-8-UT-6-4 Cajon ultra-torr fitting #SS-16-UT-A-20 Cajon ultra-torr union fitting 1/4" to 1/4" #SS-4-UT-6 Cajon tube adapter NPT Male #B-2-MHC-4S for water bath Nupro air actuator valve #SS4-BK-1C Nupro DC powered soledoid valve #MS-sol-1K Cajon flexible tubing 12" #321-4-X-12 Cajon flexible tubing insert #304-4-XBA Cajon female 1/4" NPT Tee #B-4-T Cajon 1/4" NPT male x 1/4" swagelok #B-MB4-TA-1-4-24R Swagelok x 10/32" male #B-400-1-0232
Cole Parmer 7425 N. Oak Park Ave. Chicago, IL 60648	3/8" OD x 1/4" ID Norprene tubing #N-06410-05 Plastic compression quick-disconnects #YA-6360-30 & #YA-6364-05 Finnpippette 5-50ul #J-6247-01
General Supply & Metals 47 Nauset Street New Bedford, MA 02746	1/4" OD copper rod (sample boats; see appendix 4.)
Hallmark Electronics 6 Cook St. Pinehurst Park Billerica, MA 01821	8K Non-volitile ram chips #DS1225Y
Lambda Electronics 515 Broad Hollow Rd. Melville, NY 11746	Power supply 0-40 VDC #LQ-412
Matrix Corp. 1203 New Hope Rd. Raleigh, NC 27610	Stepping Motor control card for STD bus #7911/SSC-5K Unipolar motor driver card #7911/usd RCA/26 cable to connect SSC to USD
Mettler Inst. Corp. Box 71 Hightstown, NJ 08520	#012 bi-directional data interface for AE balance
Microgroup 7 Industrial Park Rd. Medway, MA 02053	304 low carbon seamless s.s. tubing 1/4" OD x .060" ID x .095" wall Military specs .021% carbon
MKS Instruments 6 Shattuck Rd. Andover, MA 01810	Absolute pressure transducer #122AA-00100-AB 0-100 Torr; 1/2" tubulation with standard accuracy plus cable and connector
Newark Electronics Route 1 South Park Walpole, MA 02081	26 pin female edge connector assembly #FCE-26-103 26 pin flat cable connector with polarity bump #S2J026 26 conductor flat laminated cable #36F658WA
Poly Sciences Corp. 7800 Merrimac Ave. Niles, IL 60648-48312	Circulating water bath #1-060-700
Versalogic 3888 Stewart Road Eugene, OR 97402	64K ram memory expansion board #VL-7709A Z-80 processor card for STD bus #VL-7806C C4 Basic/Novos ROM for VL-7806 #2066 STD-Ain-1A analog input card #2370 Power supply card #VL-PSC 8 channel relay output card #VL-IPI-2 Transformer #PSC-2532 9 slot card cage with backplate #VX-09T-MB #8560 26 pin to D8-25S cable #9553 D8-25S to D8-25S cable (for serial connection)
VWR Scientific P.O. Box 232 Boston, MA 02101	Magnetic stirrer #58940-158 1/4" x 7/8" glass stir bar #58948-422 Labindustries repipete, 20 ml #53527-121

Appendix 9. Trouble shooting guide

TROUBLE	POSSIBLE CAUSE
Carousel jams	The carousel spindle must be absolutly perpendicular to the reducing gears. Try using ball bearings of a slightly different size or number below the carousel. A copper disk can also be placed into the depression in the lower carousel assembly to change the height of the ball bearings. The two reducing gears on the carousel spindle are not perfectly aligned. Trial and error will locate the correct allignment.
Balance won't tare	Try reinitializing the memory first. If the balance still refuses to tare, quit the Automated Carbonate System program and rerun the machine language loader program followed by the ACS program. The balance will now have no trouble taring.
Carousel doesn't stay aligned to hole one	The gears are probably slipping. Tighten.
Standards are low	There is water in the standards. Bake. Standards are being sucked away during the initial pumpdown. Standards 35-5 will be most affected. Try adding more methanol to the boats before running. The acid bath temperature is too low causing CO_2 to remain dissolved in the acid. Raise the water bath temperature.
Standards are high or erratic	Check the regression line by producing a run of standards. The pressure transducer does get dirty and needs to be sent to the factory once a year for cleaning and recalibration. Check to make sure the temperature of the water bath is at the correct setting.
Incomplete reactions	If more than one sample per run does not react fully during the 5 minute reaction time, choose a longer reaction time. Raise the temperature of the water bath to reduce the viscosity of the acid. Increase the stir bar speed.
Acid is thick at the end of the run	Low carbonate samples with lots of clay were probably run and larger weights were used to compensate for the smaller pressure expected. Lower the sample weights to the same range as used for higher carbonate samples.

Appendix 10. System startup.

The following directions allow the user to operate the carbonate system quickly and easily. For more detailed information, refer to the technical manual.

1. Make sure the compressed air source is on and adjusted to 80 psi.
2. Turn on the stepping motor power supply making sure either the (+) or the (–) terminal lead is *not* connected.
3. Once the voltage stabilizes at 28VDC, plug in the unattached lead. If there has been a power failure, the surge protection strip will probably have to be reset.
4. Power up the **NEC** terminal. *ASYNC* should be installed in drive 'A'.
5. Power up the card cage and then push the reset button.
6. The *C4 BASIC* system prompt '*' should be displayed on the screen. If not, push the reset button on the card cage.
7. Type 'DIR' at the prompt to make sure the following is printed:
   1. The Animating Elememt, last revised 30.08.89 (DRO)
   2. RDKEY & METIO loader program, last revised 29.08.89 (DRO)
   3. GETDATA program to produce pressure/time data
8. Type 'run 2 <RET>' (nothing will appear to happen) followed by 'run 1 <RET>'.
   The following should appear:
   Welcome stranger.
   Skip software initialization? (Y/N) : (in most cases type 'N')
   Is the lazy susan homed to slot #1? (Y/N) : (check the alignment)
9. 'The Main Menu' will now be displayed on the screen.
10. Attach the acid filled reaction vessel to the carousel.
11. Turn the circulating water bath on, checking that it is adjusted to 80°C.
12. Turn on the spin bar mixer to a setting of 3.

Appendix 10. System startup continued.

13. Make sure the **ULTRA-TORR** fitting on the carousel is tightened.
14. Turn on the vacuum pump with the toggle switch on the surge protection strip.
15. Go to the ‘Tweak the System Menu’ and hit ‘V’ for vacuum. The acid will now begin to outgas as it is heated to temperature under vacuum.
16. Weigh out your dried/crushed sediment samples. Add methanol to each sample and dry in the oven.
17. Load the samples into the carousel. Tighten the **ULTRA-TORR** fitting.
18. Run the samples, choosing the appropriate reaction time.
19. At the end of the run, ‘The Main Menu’ will be printed on the screen. Make sure there is a floppy disk inserted in drive ‘B’.
20. Type ‘^ VF <RET>’ : (‘File specification:’ will appear on the screen)
21. Type ‘B:filename.crb <RET>’ : (The file will be copied to drive ‘B’)
22. Type ‘^ VG <RET>’ : (‘port open’ will appear on the screen)
23. Hit ‘P’ and the contents of the carbonate system memory will be printed to the screen one line at a time.
24. When all 40 lines of data have been printed, type ‘^ Z <RET>’.

DOCUMENT LIBRARY

January 17, 1990

Distribution List for Technical Report Exchange

Attn: Stella Sanchez-Wade
Documents Section
Scripps Institution of Oceanography
Library, Mail Code C-075C
La Jolla, CA 92093

Hancock Library of Biology &
Oceanography
Alan Hancock Laboratory
University of Southern California
University Park
Los Angeles, CA 90089-0371

Gifts & Exchanges
Library
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, NS, B2Y 4A2, CANADA

Office of the International
Ice Patrol
c/o Coast Guard R & D Center
Avery Point
Groton, CT 06340

NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL 33149

Library
Skidaway Institute of Oceanography
P.O. Box 13687
Savannah, GA 31416

Institute of Geophysics
University of Hawaii
Library Room 252
2525 Correa Road
Honolulu, HI 96822

Marine Resources Information Center
Building E38-320
MIT
Cambridge, MA 02139

Library
Lamont-Doherty Geological
Observatory
Columbia University
Palisades, NY 10964

Library
Serials Department
Oregon State University
Corvallis, OR 97331

Pell Marine Science Library
University of Rhode Island
Narragansett Bay Campus
Narragansett, RI 02882

Working Collection
Texas A&M University
Dept. of Oceanography
College Station, TX 77843

Library
Virginia Institute of Marine Science
Gloucester Point, VA 23062

Fisheries-Oceanography Library
151 Oceanography Teaching Bldg.
University of Washington
Seattle, WA 98195

Library
R.S.M.A.S.
University of Miami
4600 Rickenbacker Causeway
Miami, FL 33149

Maury Oceanographic Library
Naval Oceanographic Office
Bay St. Louis
NSTL, MS 39522-5001

Marine Sciences Collection
Mayaguez Campus Library
University of Puerto Rico
Mayagues, Puerto Rico 00708

Library
Institute of Oceanographic Sciences
Deacon Laboratory
Wormley, Godalming
Surrey GU8 5UB
UNITED KINGDOM

The Librarian
CSIRO Marine Laboratories
G.P.O. Box 1538
Hobart, Tasmania
AUSTRALIA 7001

Library
Proudman Oceanographic Laboratory
Bidston Observatory
Birkenhead
Merseyside L43 7 RA
UNITED KINGDOM

Mac 90-32

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-90-03	2.	3. Recipient's Accession No.

4. Title and Subtitle Automated System to Measure the Carbonate Concentration of Sediments	5. Report Date January, 1990 6.
7. Author(s) Dorinda R. Ostermann, Darrell Karbott and W.B. Curry	8. Performing Organization Rept. No. WHOI 90-03
9. Performing Organization Name and Address The Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543	10. Project/Task/Work Unit No. 11. Contract(C) or Grant(G) No. (C) OCE85-11014 (G) OCE88-13307
12. Sponsoring Organization Name and Address Funding was provided by the National Science Foundation	13. Type of Report & Period Covered Technical Report 14.

15. Supplementary Notes

This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept., WHOI-90-03.

16. Abstract (Limit: 200 words)

We have developed a computer controlled system to measure the calcium carbonate content of sediment samples. A menu driven program controls the analysis of each sample. The system first communicates with a Mettler digital balance to record the weights of the 40 samples which must be loaded into each run. The sample boats are next loaded into the sample carousel which is then sealed from the atmosphere. The system is first pumped down to a vacuum of 0.04 torr. The valve to the pump closes and the stepping motor turns the carousel, moving a sample boat over the delivery slot and dropping the sample into 80°C 100% phosphoric acid under vigorous spinning action. During the reaction, carbonate is evolved into H_20 and CO_2 and the resulting pressure change within the closed system is measured by a pressure transducer and recorded into memory next to the sample identification and sample weight. The system is pumped once again to 0.04 torr and the process continues until all 40 samples have been analyzed. The data can then be uploaded and converted to percent carbonate values using a regression line produced from multiple analyses of varying weights of a 100% carbonate standard. Precision of the system, based upon 120 replicate analysis ranges from 0.49% to 0.88%.

17. Document Analysis a. Descriptors

1. carbonate
2. automated
3. sediments

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for publication; distribution unlimited.	UNCLASSIFIED	17
	20. Security Class (This Page)	22. Price

(See ANSI-Z39.18) *See Instructions on Reverse* OPTIONAL FORM 272 (4-77) (Formerly NTIS-35) Department of Commerce