WHOI-94-01

Copy 2

Woods Hole Oceanographic Institution

1930

MEASURING xCO_2 USING THE CAT/NDIR METHOD
System set-up, calibration, maintenance and shutdown

by

Maren E. Tracy, Edward T. Peltzer and Catherine Goyet

February 1994

Technical Report

Funding was provided by the Department of Energy under Grant No. FG02 94ER61544.

WHOI-94-01

MEASURING xCO_2 USING THE CAT/NDIR METHOD
System set-up, calibration, maintenance and shutdown

by

Maren E. Tracy, Edward T. Peltzer and Catherine Goyet

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

February 1994

Technical Report

Funding was provided by the Department of Energy under Grant No. FG02 94ER61544.

Approved for public release; distribution unlimited.

Approved for Distribution:

Geoffrey Thompson, Chair
Department of Marine Chemistry and Geochemistry

Table of contents

Introduction:

The purpose of this document is to provide the operator of the CAT/NDIR system a guidline for setting up or shutting down the system in addition to daily and weekly maintenance.

Principle of xCO_2 measurements using a CAT/NDIR system:

The automated underway xCO_2 monitoring system consists of a "shower-head" type equilibrator and a non-dispersive infra-red (NDIR) analyzer (Li-COR model 6262) with solid state detector. A system of automated valves (Figure 1) controls the frequent and regular switching of gas flow to the NDIR analyzer between seawater equilibrated air (SEA), seasurface air sampled at the ship
bow (AIR), and two gas standards (high and low CO_2 concentration, respectively). The small size (approximately 40 cm high) equilibrator (modified from that used by Weiss) consists of two concentric cylindrical stages constructed of plexiglass, with a drain in the center. The seawater "showers" through the top of the equilibrator, and the first stage of the equilibrator is vented to the clean marine atmosphere to maintain ambient pressure. The gas phase is continuously re-circulated, at a rate of 200 ml/minute, by an air pump, through a closed loop passing through the infra-red analyzer where the measurement is made. The seawater temperature in the equilibrator, as well as both atmospheric pressure and the gas pressure in the closed loop are continously monitored.
The CO_2/H_2O differential NDIR analyzer is of small size, precise, and insensitive to vibrations and lateral accelerations. The sample cells are gold-plated to enhance IR reflectivity and resist tarnishing over time. One set of cells is used for both H_2O and CO_2 measurements by using a dichroic beam splitter to provide radiation to two separate detectors. A 150 nm bandpass optical filter is used to select the 4.26 micron absorption band for CO_2 detection, and the H_2O detector is filtered for the 2.59 micron absorption band. Both filters provide excellent rejection of IR radiation outside the desired band, allowing the analyzer to reject the response of other IR absorbing gases. The filters are mounted directly on the detectors for thermal stability. The lead selenide solid state detectors are cooled and regulated at -12°C by thermoelectric coolers, and electronic circuits continuously monitor and maintain a constant detector sensitivity. The detector housing is maintained free of water vapor and CO_2 by internally mounted desiccant and absorbants. In order to maximize the signal sensitivity, the infra-red radiation from the source is focused through the gas cell and onto the detector by lenses at each end of the optical bench. As

a result, the typical CO_2 noise level is 0.2 ppm peak-to-peak (at 350 ppm) when using one second signal averaging. This automated system allows us to monitor xCO_2 (mole fraction) directly in the gas phase without having to pretreat it (no drying or gas separation are required).

f_R = filter; gelman 1.0 μ; reference
f_S = filter; gelman 1.0 μ; sample
V_1, V_3, V_5, V_7 = 3-way electrical valves
PT_1 = Pressure Transducer (line pressure)
PT_2 = Pressure Transducer (ambient)
RM_R = rotometer; reference
RM_S = rotometer; sample
T_R = trap; ref. (soda lime & $Mg(ClO_4)_2$)
T_S = trap; sample (soda lime & $Mg(ClO_4)_2$)
P_A = pump, Air
P_R = pump; reference
P_S = pump; sample
^3V_A, ^3V_B = 3-way valves
X_1, X_2, X_3, X_4 = regulating valves
D.P.G. = Dew Point Generator
7M = 7μ filter

Tubing legend

- 3/8" Dekoran type 1300
- 1/8" copper
- 1/4" copper
- 1/8" Bev-a-line

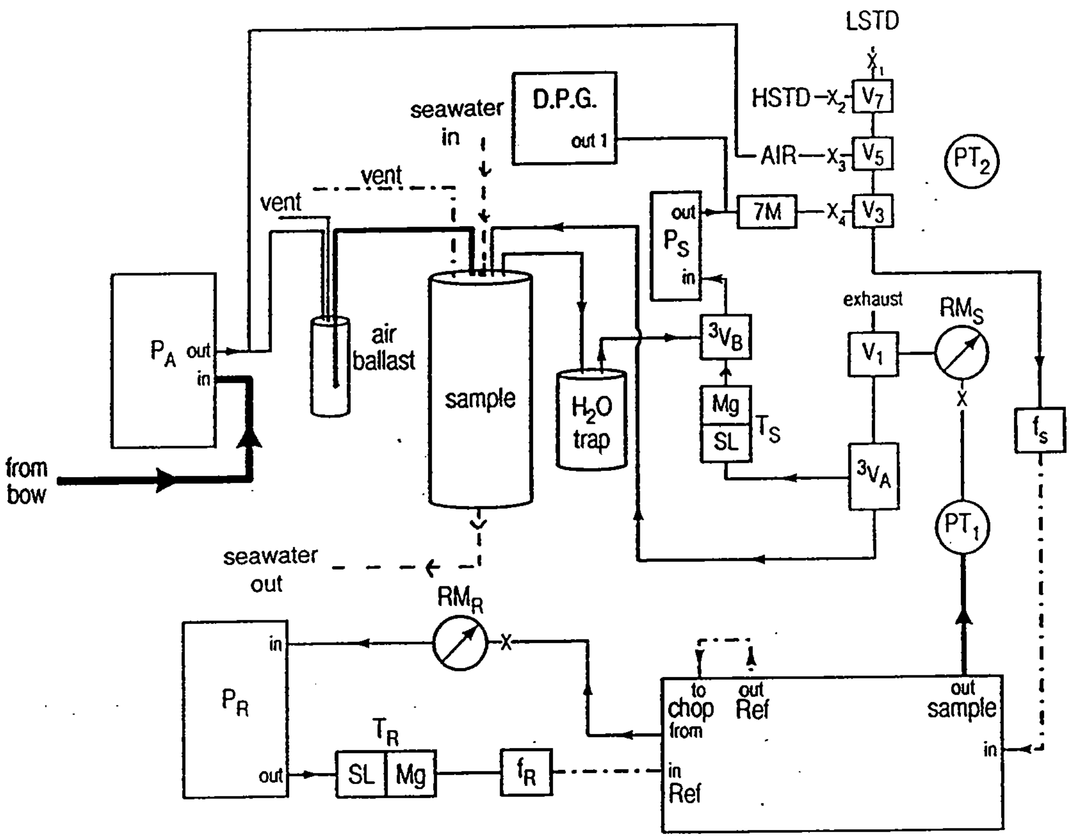


Figure 1.

$\underline{C}O_2$ Automated Technique (CAT)

Continous underway xCO_2 monitoring system

This not only simplifies the measurement procedure, but also minimizes the potential errors in the measurements. The Li-Cor CO_2 /H_2O analyzer uses an internal algorithm to correct the measurements to a dry gas scale.

This system is regularly calibrated every two hours, using CO_2 standard gases calibrated against those from the National Institute of Standards and Technology (NIST). The recorded xCO_2 data consist of an average of 10 readings taken every 6 seconds. The rate of data storage is one measurement per minute. A typical duty cycle consists of a five-minute cell flush with seawater equilibrated air, followed by five one-minute averages. Next, this cycle is repeated for the measurement of air (the air intake is located at the bow of the ship). Every two hours the alternation of these two sample gases is interrupted with the measurement of the two reference gases according to the same ten minute cycle.

(The technical description of the LI-COR 6262 operation is taken from the LI-COR UpDate, Vol.4,#1,pp1,2)

Set up and installation of the system:

Setting up the CAT (CO2 Automated Technique), can be conveniently broken down into 6 areas: Sample collection tubing installation (Dekoron TM), equilibrator installation, computer installation, gas installation, Dew Point Generator (D.P.G.) installation and NDIR (Non-Dispersive Infra-Red) analyser (LI-COR LI-6262) calibration. The CAT is self-contained in its own shipping container.

Dekobon Installation:

In order to analyse the air sample we need to collect it. The CAT is a continually monitoring apparatus. The air sample is collected from the bow or stern depending on the ship's orientaion relative to the wind direction, and pumped to the CAT through clean Dekobon tubing. Dekobon is the registered trademark of Furon Dekoron Co.. We use Dekabon,Type 1300 metal/plastic composite tubing with 3/8" O.D.. The tubing is supplied clean from the manufacturer and no further cleaning is recommended. The intake end of the tubing should be positioned at a level well above the deck of the ship where any contamination from the ship's exhaust or personnel will be minimized. Generally a good place to put the intake is 3-5 meters above the deck on the jack staff or other forward positioned mast or staff. A funnel should be attached, inverted, to the intake end of the tubing, (secured with a banding clamp and sealed with silicone) to prevent collection of seawater spray or rainwater. Mount the Dekoron tubing at the correct height , and secure the tubing along the side of the ship and the mast with cable ties every meter. *Before going aloft be sure to get permission from the ship's watch. Use appropriate safety equipment and alert the appropriate watch personel.* Run the tubing the length of the vessel to the lab where analysis will be performed. Be careful to install the tubing in low traffic areas, if possible, to eliminate accidental crimping or cutting of the tubing. If the tubing must be installed in high traffic areas, make sure it is well protected and inspect it daily throughout the cruise for possible damage. The Dekobon tubing is terminated at the water trap before the sample is pumped through the air pump (P_A). The output of P_A is split with a tee. One path is to the air input (V_5) of the CAT and the other path is to the air ballast and then into the equilibrator.

Equilibrator installation:

The equilibrator must have an uncontaminated seawater source. The seawater input will optimally have a remote on/off valve. An empirical experiment in the laboratory suggests that the equilibrator works best when seawater flows at a rate of 4 liters per minute. The flow can be controlled and monitored by a combination flow meter and valve or if the line is dedicated, the flow can be set initially and regularly checked every few days. If the consistancy of the flow is variable, a metering device in line, is highly recommended. An overflow of seawater into the system will force water into the NDIR, corroding the coating of the measurement cell. This should be avoided.

It is best to drain the seawater from the equilibrator overboard, (not into the ship's holding tanks). Check with the ship's engineers for the nearest overboard drain or use a garden hose to run out and overboard. Make sure the drain has a free run; the flow must be unrestricted (or again, overflowing the equilibrator and the whole system may occur).

The air input to the equilibrator from the air ballast uses 1/4" clean copper tubing. The gas sample output and the return from valve $^{3}V_{A}$ use 1/8" clean copper tubing connected with swagelok reducing unions (see Figure 2). The connection to the inner chamber is vented to clean marine air to eliminate internal pressure imbalances. The procedure for cleaning the copper tubing is described in Appendix A.

Air from the equilibrator is first pumped through a water trap mounted on the side of the equilibrator housing and subsequently through a Whatman vacu-gard filter to prevent salt water aerosols from damaging the NDIR and/or electrical valves.

Abstract

Accurate measurement of partial pressure of CO_2 in seawater is currently performed by measuring pCO_2 in an aliquot of a small volume of gas equilibrated with a large volume of the seawater to be measured. PCO_2 in the gas phase can be accurately measured either by gas chromatography or infra-red analysis. In order to minimize human labor to monitor pCO_2 in surface seawater we opted for the infra-red analysis which does not require a highly trained person and which can easily be automated. This report describes how we have designed and automated a system for continual surface seawater pCO_2 monitoring. It further indicates the necessary steps to set up, run, and maintain the system. With minors modifications this system can also be used to measure pCO_2 in discrete seawater samples (Goyet et al., 1993).

Reference:

Goyet C., Millero F.J., Poisson A., and Shafer D.K. (1993). Temperature dependence of CO_2 fugacity in seawater. Marine Chemistry, 44, 205-219.

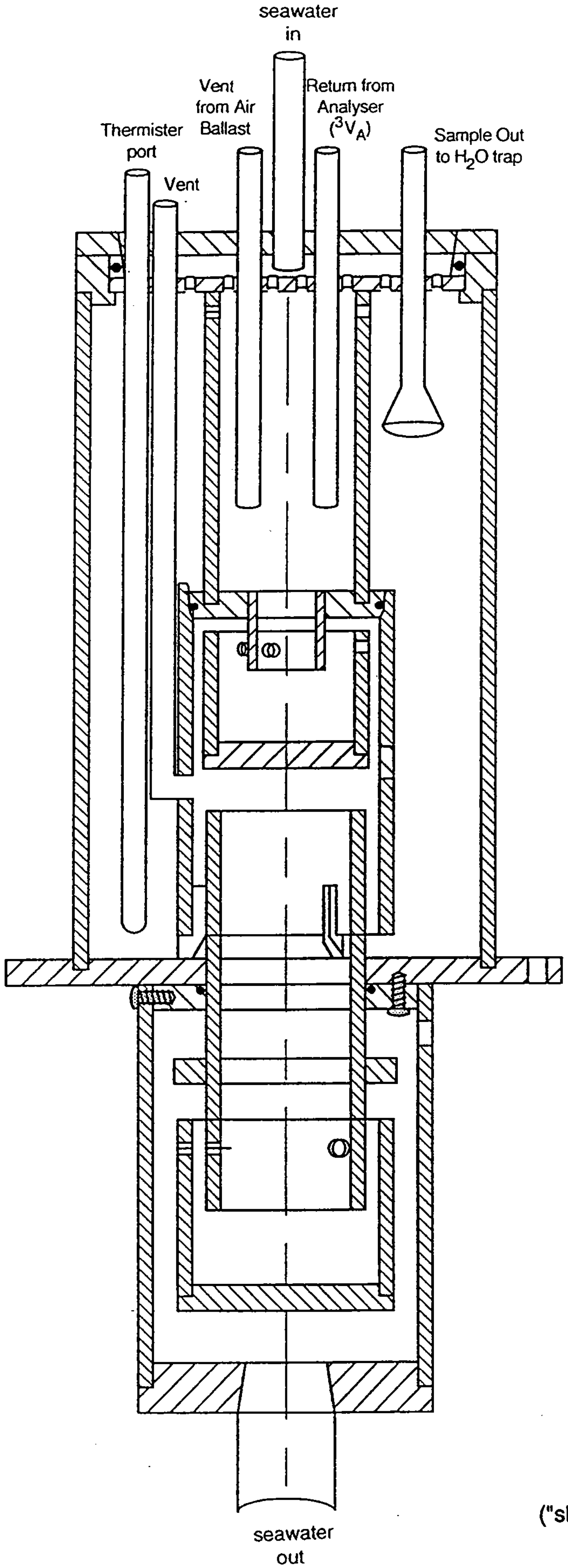


Figure 2.
Equilibrator.
("shower-head" type)

Computer Installation:

Software:

The software that drives the CAT and collects data is well documented in the Analyser Manual. It is written in Quick Basic. There are four programs:

licor.bas: This program is to be used on a computer that uses com1 to retrieve data from the LI-6262.

licor2.bas: This program is to be used on a computer that uses com2 to retrieve data from the LI-6262.

licorn.bas: This program is to be used on a computer that uses com1 to retrieve data from the LI-6262.
Additionally this program outputs CO_2 and H_2O mV readings.

The pressure coefficient and constant used in the algorithm are for the transducer pair used on xCO_2 apparatus 1 (CAT 1).

licor2n.bas: This program is to be used on a computer that uses com1 to retrieve data from the LI-6262.
Additionally this program outputs CO_2 and H_2O mV readings.

The pressure coefficient and constant used in the algorithm are for the transducer pair used on xCO_2 apparatus 2 (CAT 2).

Hardware:

The A-D converter board controlling the interface between the computer and the CAT is a Dascon1. The transducers (Setra) and thermister (YSI), outputs are input to the computer through this board, the electrical valves are also driven by the Dascon1 board.
The voltage converter on the CAT apparatus provides 24 VDC output to the Delta pumps and the transducers.
There are two cable connections; the Dascon1 ribbon cable to the "blue box" on the CAT and the connection of the LI-COR 6262 to the RS 232 input of the computer (either com1 or com2). The cable connections should be well secured.

Gas Installation:

Ideally, the CAT set up requires 4 cylinders of CO_2 in air: 2 full size working cylinders with "high" and "low" concentrations of CO_2 in air (HSTD and LSTD), spanning the anticipated measured values, and 2 sample size reference cylinders of CO_2 in air. The reference cylinders have been calibrated at W.H.O.I. against a NIST standard. They are to be used initially to set the span of the NDIR and to measure the HSTD and LSTD gases.

CO_2 in air requires a CGA 590 fitting on the regulator. The CGA 590 fitting is left hand threaded, (i.e. turn it anti-clockwise to tighten and clockwise to loosen). A two stage, metal diaphram regulator, (Linde UPE 3 25 590) is essential. The delivery stage on this regulator has a compound gauge with a range of 30 in. Hg vacuum to 30 psig. The supply stage range is 0 to 4000 psig. The regulators need to be purged three times with CO_2 cylinder gas before being connected to the CAT. 1/8" clean copper tubing is used to connect the gas cylinders to the respective CAT inputs. The delivery pressure of both HSTD and LSTD should be approximately 8 p.s.i..

For the initial calibration only the reference cylinders need to be hooked up to the CAT .

REMARKS:

BE EXTREMELY CAREFUL TO KEEP THE COPPER TUBING CLEAR OF THE ELECTRICAL CONNECTIONS.

LEAK CHECK CAREFULLY AND OFTEN.

DO NOT LET THE GAS LINES VIBRATE AGAINST EACH OTHER.

SECURE THE GAS LINES WITH CABLE TIES WHEREVER POSSIBLE.

Dew Point Generator:

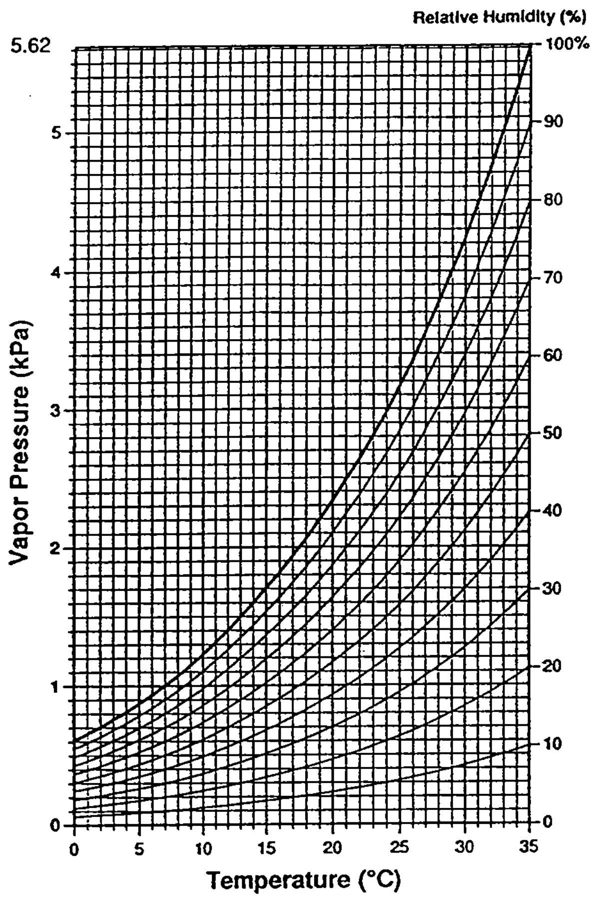
The Dew Point Generator (DPG) is used to set the span of the H_2O channel. Set up and operation are well documented in the instruction manual provided by LI-COR Inc.

Fill the radiator and condensor block with deionized water, adding algicide as suggested in the manual.

Turn on the unit and adjust flow from output 1 into the CAT by opening or restricting the flow of output 2.

Determine the DPG set point and NDIR span target using the psychrometric chart provided in the DPG manual. (Figure 3).

Psychrometric Chart (0 to 35 °C)



Saturation Vapor Pressure Table

Temp.°C	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9
0	0.614	0.618	0.623	0.627	0.632	0.636	0.641	0.646	0.650	0.655
1	0.660	0.664	0.669	0.674	0.679	0.684	0.689	0.694	0.699	0.704
2	0.709	0.714	0.719	0.724	0.729	0.734	0.740	0.745	0.750	0.756
3	0.761	0.766	0.772	0.777	0.783	0.788	0.794	0.800	0.805	0.811
4	0.817	0.822	0.828	0.834	0.840	0.846	0.852	0.858	0.864	0.870
5	0.876	0.882	0.888	0.894	0.901	0.907	0.913	0.920	0.926	0.932
6	0.939	0.945	0.952	0.958	0.965	0.972	0.979	0.985	0.992	0.999
7	1.006	1.013	1.020	1.027	1.034	1.041	1.048	1.055	1.062	1.070
8	1.077	1.084	1.092	1.099	1.107	1.114	1.122	1.129	1.137	1.145
9	1.152	1.160	1.168	1.176	1.184	1.192	1.200	1.208	1.216	1.224
10	1.233	1.241	1.249	1.258	1.266	1.275	1.283	1.292	1.300	1.309
11	1.318	1.326	1.335	1.344	1.353	1.362	1.371	1.380	1.389	1.399
12	1.408	1.417	1.427	1.436	1.445	1.455	1.465	1.474	1.484	1.494
13	1.503	1.513	1.523	1.533	1.543	1.553	1.563	1.574	1.584	1.594
14	1.605	1.615	1.626	1.636	1.647	1.657	1.668	1.679	1.690	1.701
15	1.712	1.723	1.734	1.745	1.756	1.768	1.779	1.790	1.802	1.814
16	1.825	1.837	1.849	1.860	1.872	1.884	1.896	1.908	1.921	1.933
17	1.945	1.957	1.970	1.982	1.995	2.008	2.020	2.033	2.046	2.059
18	2.072	2.085	2.098	2.111	2.125	2.138	2.151	2.165	2.179	2.192
19	2.206	2.220	2.234	2.248	2.262	2.276	2.290	2.304	2.318	2.333
20	2.347	2.362	2.377	2.391	2.406	2.421	2.436	2.451	2.466	2.482
21	2.497	2.512	2.528	2.543	2.559	2.575	2.590	2.606	2.622	2.638
22	2.654	2.671	2.687	2.703	2.720	2.737	2.753	2.770	2.787	2.804
23	2.821	2.838	2.855	2.872	2.890	2.907	2.925	2.943	2.960	2.978
24	2.996	3.014	3.032	3.051	3.069	3.087	3.106	3.124	3.143	3.162
25	3.181	3.200	3.219	3.238	3.258	3.277	3.296	3.316	3.336	3.356
26	3.376	3.396	3.416	3.436	3.456	3.477	3.497	3.518	3.539	3.559
27	3.580	3.602	3.623	3.644	3.665	3.687	3.709	3.730	3.752	3.774
28	3.796	3.818	3.841	3.863	3.886	3.908	3.931	3.954	3.977	4.000
29	4.023	4.046	4.070	4.093	4.117	4.141	4.165	4.189	4.213	4.237
30	4.262	4.286	4.311	4.336	4.361	4.386	4.411	4.436	4.461	4.487
31	4.513	4.538	4.564	4.590	4.617	4.643	4.669	4.696	4.722	4.749
32	4.776	4.803	4.831	4.858	4.885	4.913	4.941	4.969	4.997	5.025
33	5.053	5.082	5.110	5.139	5.168	5.197	5.226	5.255	5.285	5.314
34	5.344	5.374	5.404	5.434	5.464	5.495	5.525	5.556	5.587	5.618
35	5.649	5.680	5.712	5.743	5.775	5.807	5.839	5.871	5.904	5.936
36	5.969	6.002	6.035	6.068	6.101	6.135	6.168	6.202	6.236	6.270
37	6.305	6.339	6.374	6.409	6.443	6.479	6.514	6.549	6.585	6.621
38	6.657	6.693	6.729	6.765	6.802	6.839	6.876	6.913	6.950	6.988
39	7.025	7.063	7.101	7.139	7.178	7.216	7.255	7.294	7.333	7.372
40	7.412	7.451	7.491	7.531	7.571	7.612	7.652	7.693	7.734	7.775
41	7.816	7.858	7.899	7.941	7.983	8.025	8.068	8.110	8.153	8.196
42	8.239	8.283	8.327	8.370	8.414	8.459	8.503	8.548	8.592	8.637
43	8.683	8.728	8.774	8.819	8.865	8.912	8.958	9.005	9.052	9.099
44	9.146	9.193	9.241	9.289	9.337	9.386	9.434	9.483	9.532	9.581
45	9.631	9.680	9.730	9.780	9.830	9.881	9.932	9.983	10.034	10.085
46	10.137	10.189	10.241	10.293	10.346	10.399	10.452	10.505	10.559	10.612
47	10.666	10.720	10.775	10.830	10.885	10.940	10.995	11.051	11.107	11.163
48	11.219	11.276	11.333	11.390	11.447	11.505	11.562	11.621	11.679	11.737
49	11.796	11.855	11.915	11.974	12.034	12.094	12.155	12.215	12.276	12.337
50	12.399	12.461	12.523	12.585	12.647	12.710	12.773	12.836	12.900	12.964

This table gives the saturation vapor pressure of moist air in kPa. This table was derived using the following equation from Buck (1981):

$$e(T) = 0.61121\left[1.00072 + 3.2\times10^{-5}(P) + 5.9\times10^{-9}(PT)\right]e^{\frac{\left(18.729-\frac{T}{227.3}\right)T}{T+257.87}}$$

Figure 3.
Psychrometric chart and saturation vapor table for use calibrating the Dew Point Generator (Reproduced here from the Li-610 Instruction manual, page C-2, with permission of Li-Cor inc.).

NDIR/ CAT Set up and calibration:

The NDIR is mounted on an acrylic base with the associated valves, pumps and pressure transducers, etc. The NDIR /CAT samples the gases as described in the section Method of Analysis. There are two loops being sampled; the reference loop and the sample loop. The reference loop is a closed loop, consisting of a Delta pump which moves gas out of the analyzer and through a roto-meter at 200 ml/min.. The flow is controlled by adjustment of a potentiometer on the pump. CO_2 and H_2O are removed from the gas stream by a desiccant/absorbant trap containing soda lime and magnesium perchlorate ($Mg(ClO_4)_2$), followed by a gelman 1 μm filter which removes any particles.

The sample loop is more complex. A Delta or metal bellows pump is used to move the gas stream at 200 ml/min. The sample flows out of the NDIR, into a pressure transducer and through a roto-meter. At this point the flow path diverges and is dependent on the requirements of the operation. For determining the NDIR zero set point, the three way valves ^3V_A and ^3V_B are adjusted allowing the gas to flow through a trap/filter configuation identical to the reference loop. Determining the span set point of the NDIR requires the valves ^3V_A and ^3V_B be returned to the sampling position and electrical valves opened via computer command, sampling the HSTD/LSTD gases consecutively. Sampling the AIR and SEA gases is accomplished automatically as the appropriate valves are opened by the computer program.

The desiccant/absorbant traps should be installed in the respective reference and sample loops. The traps are filled with 1/2 soda lime (4-8 mesh) and 1/2 $Mg(ClO_4)_2$ (12-20 mesh). Appropriate care should be used when handling these chemicals. The chemicals in the traps are separated by phosphoric acid washed glass wool. Orientation of the traps in the CAT is critical, as the soda lime removes CO_2 and requires a small amount of H_2O to be effective, therefore the gas must first flow through the soda-lime and then through the $Mg(ClO_4)_2$. The correct orientation is indicated in the gas flow diagram (figure 1).

Connections to the LSTD, HSTD, SEA, and AIR are made using swagelok fittings. Gas flow rates are precisely controlled at the input to the CAT using needle valves (X_1 -X_4). Flow rates are 200 ml/min. for the sample and reference lines. All connections should be carefully leak checked.

Calibration:

Once the integrity of gas input and output connections to the CAT are assured, the system is ready to be calibrated as follows:

Turn on the auxillary power to the CAT (switch on the yellow cord).

Turn on the computer.

Turn on the NDIR.

Set Date/Time:

Set the computer clock to the correct date/time using UTC (Universal Time Coordinate). It is optimal to sychronize the computer clock with the ship's navigation logging system.

Data Collection:

Make a subdirectory on the hard drive within \pco2 in which data will be stored as it is collected.

Example: Data from the WOCE P17 cruise may be stored in subdirectory c:\pco2\wocep17.

Change directory to c:\pco2.

Start program operation by typing the appropriate program name.

For example: If you are using xCO_2 apparatus 1 (CAT 1) and the NDIR is connected to the computer via com1, the correct program for collecting data is licorn.bas and you simply type **<licorn> <enter>** at the prompt (c:\pco2>).

Begin calibration:

The reference cylinders must be hooked up to the HSTD/LSTD inputs.

Turn the three way valves to allow flow through the desiccant/absorbant trap.

Choose option 5 from the main menu and follow the directions.

Allow approximately 15-20 minutes for stability.

Adjust the CO_2 and H_2O zero pots on the NDIR .

Check for stability.

<Ctrl Enter> to step through the calibration procedure.

Disconnect the reference cylinders and hookup the HSTD snd LSTD cylinders.

Disconnect and turn off the Dew Point Generator.

Leak check the system. (If any leaks are found you may need to re-calibrate using the reference cylinders).

Measure the CO_2 concentration of the HSTD and LSTD gases and mark the values on the cylinders.

Begin data collection.

Choose option 6 from the main menu and type in the correct path and filename.

For example: Using the subdirectory created during the computer setup and the current date, the path and filename will be **wocep17\930326 <enter>.**

Remember to use UTC.

Begin sample analysis.

Choose option 9 from the main menu.

Choose option 9 from the sub-menu.

Make sure the data collection has begun .

Make sure UTC is used.

Maintenance:

Daily Maintenance:

Close existing file and open a new file:
In order to maintain files of a resonable size and to insure accurate record keeping, every 24 hours the data collection file is closed and reopened with the new date/time.

- **<Ctrl enter>** to halt program
- Choose option 7 to close file
- Choose option 6 to open new file using subdirectory,filename (yymmdd);
  \wocep17\930428 <enter>
- Choose option 9 to restart data collection

Check visually for any water in the trap, empty daily if required.

Check the vacugard filter in line from the equilbrator/water trap:

- If the filter shows any indication of water in the filter housing, replace the filter with a new one, taking care to orient it correctly.

Monitor CAT/NDIR operation on restart and note HSTD, LSTD,SEA and EQUIL readings in the notebook.
Also note weather conditions, ships position and ship's operation status.
Check for correct gas flows in both reference and sample loops (200 ml/min.).
Check seawater flow, should be approximately 4 liters per minute.
Check equilibrator drain, should be free flowing.
Check the water filter and change when necessary (every 2-3 days).

- If the filter is not wetted completely a problem may develop, as air can become trapped in the lines restricting the seawater flow.
- Install the new filter in the filter cartridge holder. If the filter cartridge does not fill completely, let the water flow through the filter for several minutes. Turn off the water and reverse the orientation of the filter in the cartridge holder.

Weekly Maintenance:

Check visually the Gelman Acro filters in the reference loops for any contaiminants or particles.

- Replace if necessary.
- Back up data to floppy disks.

Monthly Maintenance:

Change absorbant and desiccant in reference line.

- **<Ctrl enter>** to halt data collection.
- Turn off power to Dascon1 board and reference pump ("yellow cord").
- Remove dessicant/absorbant trap and replace soda lime and magnesium perchlorate.
- The traps are filled with 1/2 soda lime and 1/2 magnesium perchlorate, separated with glass wool. Use proper care when handling these chemicals. Replace the desiccant/ absorbant tube in the CAT, observing the proper orientation.
- Turn on power to Dascon1 board and reference pump.
- Restart data collection.

In the event of a power outage:
The system is protected by a UPS system that will provide uninterupted power for approximately 30 minutes. This provides ample time to allow you to shut down the system gracefully, avoiding the potential for damage to the equipment.

Power up sequence:

- CAT on (power to Dascon1 board).
- Computer on; change directory to pco2 and choose correct program (see Measuring xCO_2 document).
- NDIR on

Shutting down the system:

- Stop data collection; Opt. 7
- Exit Program; Opt. 0
- Close the cylinder valves on the HSTD and LSTD gases.
- Turn off power to intake SEA pump.
- Close uncontaiminated seawater input valve.
- Stow drain hose so that it is out of way while in port.
- Clean equilibrator if neccessary.

Backup the data you have collected onto a floppy.

DO NOT ERASE FROM THE HARDDRIVE.

Turn off computer.
Turn off power to Dascon1 board.
Turn off the NDIR.

Please keep a copy of the data and send one copy to:

Dr. Catherine Goyet
Woods Hole Oceanographic Institution
Dept. of Marine Chemistry and Geochemistry
Woods Hole, MA 02543

(508)457-2000 ext 2552

In Case of Emergency:

Catherine Goyet

cgoyet@whoi.edu

APPENDIX A

Cleaning copper tubing.

Set up a vacumm pump with a rubber stopper in a heavy walled erlenmeyer flask with a side arm. An 1/8" copper tube terminated on one end with a female swagelock fitting should be inserted into the rubber stopper. Terminate one end of the roll of tubing to be cleaned with a female swagelock fitting and attach it with a union to the copper tubing in the stopper.

Pour ~200 ml of acetone into a glass beaker.
Pull the liquid through the tubing (200ml/50 ft.).
Allow all of the acetone to be pulled out of the tubing.
Repeat the above procedure using methylene chloride.
Put the tubing into an oven at 90°C and continue to pull air through the tubing for approximately 1 hour.

USE APPROPRIATE SAFETY PROCEDURES WHILE HANDLING AND DISPOSING OF THE SOLVENTS.

Appendix B

Data Analysis

This document is an attempt to describe and clarify the process developed for analysing data collected using the CAT/NDIR system.
Mike Hamilton of JPL developed the IMSL/IDL programs this analytical process is based upon and his contribution is greatly appreciated.

This document is divided into the following sections:

I. xCO2 data
- a.) Data collection
- b.) Data integrity
- c.) Data averaging and correction
- d.) IMSL/IDL

II. DAS data (ship's data sometimes called SAIL)
- a.) Data collection
- b.) Data integrity
- c.) Data averaging and correcting
- d.) IMSL/IDL

III. Merge the xCO2 and DAS data
- a.) xCO2 data
- b.) DAS data
- c.) IMSL/IDL

IV. Transforming xCO2 data to pCO2 data
- a.) Transforming xCO2(air) to pCO2(air)
- b.) Transforming xCO2(eq) to pCO2(eq)
- c.) Transforming pCO2(eq) to pCO2 (seawater-insitu)

V. Misc.
- a.) UNIGRAPH
- b.) Final data - headers
- c.) Final data - read.me file

NOTE: Before beginning the data analysis it is suggested that you create a new directory and move the IMSL/IDL programs as well as your data to that new directory.

I. xCO2 data

a.) Data collection:

The xCO2 data is collected using the CAT/NDIR system and stored to either the floppy or hard drive of the computer driving the system. The data is in ASCII. It is preferable to have the operator of the CAT/NDIR system stop and start the data collection regularly every 24 hours.

b.) Data integrity:

Once the data has been collected, a visual inspection is suggested.

c.) Data averaging and correction:

During this step the data will be averaged to a value representing 5 minutes of sampling. The data will also be corrected to the standard values measured during the calibration of the system.

The inspected files should be listed to a file named pco2.lis.

eos% **ls 940107 >pco2.lis**

Update the IMSL/IDL program wpco2.pro using the correct high standard reference gas value and the correct low standard reference gas value.

example:

wpco2.pro
lines 32 and 33 read:

```
a=(408.72-278.69)/(hstd-lstd)          ; compute coefficients
b=408.72-(a*hstd)
```

-the measured high standard and low standard values determined after the span of the NDIR was set should be inserted instead of 408.72 and 278.69 respectively.

Also edit the program to reflect the correct year in the reftime statement.

NOTE: An earlier version of pco2.pro was used to analyse the JGOFS data. That version does not account for the collection of mV readings and will not work on data collected after Debbie's most recent update of the basic program for the CAT/NDIR (approximately Jan.1, 1993).

d.) Run the IMSL/IDL program:

To initiate IMSL/IDL:

eos% **source imslidl_setup**	**<enter>**	"sources" the setup file
eos% **imslidl**	**<enter>**	starts IMSL/IDL

Run the program wpco2.pro to correct and average the xco2 data. The function avg.pro is called in the program and must be in the working directory. Once you have initiated IMSL/IDL issue the following commands:

==>**.run,wpco2.pro**	**<enter>**	compiles the program wpco2.pro
==>**wpco2**	**<enter>**	runs the program

(see the EOS SCF sw_packages document in /eos/bin/admin/docs for more information on IMSL/IDL)

II. DAS data

a.) Data collection:

The DAS or SAIL data is collected during the cruise by the ship's technicians.

b.) Data integrity:

Visually check the data.

c.) Run the IMSL/IDL programs:

The program wstrip.pro will strip the end characters and bad data from the DAS files.

==>**.run,wstrip.pro**	**<enter>**	compiles the program
==>**wstrip**	**<enter>**	runs the program

This program will append a .strip extension on the end of the filename. Rename the files to eliminate the extension(s).

The program dasplotw.pro will truncate any data lines with errors and then average the DAS data to 5 minute periods. The function bandav.pro is called in the program and must be in the working directory.

Depending on the format of the DAS or SAIL data, dasplot.pro may require editing...check that the dataline commands are correct.

Once you have initiated IMSL/IDL issue the following commands:

==>**.run,dasplotw.pro**	**<enter>**	compiles the program
==>**dasplotw,'filename'**	**<enter>**	runs the program

III. Merge the xCO2 and DAS data:

a.) xCO2 data

Inspect the xCO2 data, check to see that the calculations were done correctly and that the data points are valid.

List all of the corrected xCO2 data files (.corr) to pco2.lis...Be sure to erase the original pco2.lis.

eos% **ls *.corr >pco2.lis**

b.) DAS data

Inspect the .fmt files to see that the data has not been corrupted and that there are no illegal characters in the file.

List all of the formatted DAS files (filename.fmt) to a file named das.dat.

eos% **ls *.fmt >das.dat**

c.) Run the IMSL/IDL program:

Once you have initiated IMSL/IDL issue the following commands:

==>.run,mer.pro <enter> compiles the program
==>mer <enter> runs the program

IV. Transforming xCO2 data to pCO2 data:

a.) Transforming xCO2(air) to pCO2(air):

Use the measured atmospheric pressure and the measured in-situ temperature to calculate pCO2(air).

Calculate e(air):
e(air)=0.981*exp(27.029-(0.0098*(insitemp+273.15))-(6163/(insitemp+273.15)))

Calculate pCO2(air):
pCO2(air)=xCO2(air)*(airpress*10)*(1-(e(air)/(airpress*10)))/1013.25

b.) Transforming xCO2(eq) to pCO2(eq):

Use the measured atmospheric pressure and the measured temperature in the equilibrator to calculate pCO2(eq).

Calculate e(eq):
e(eq)=0.981*exp(27.029-(0.0098*(eqtemp+273.15))-(6163/(eqtemp+273.15)))

Calculate pCO2(eq):
pCO2(eq)=xCO2(sea)*(seapress*10)*(1-(e(eq)/(seapress*10)))/1013.25

c.) Transforming pCO2(eq) to pCO2 (seawater-insitu):

No comment, I've never performed this step.

V. Misc.

a.) UNIGRAPH

I use UNIGRAPH to do most of the calculations and all of the plotting. I created executable .ugr files (in directory UNIGRAPH). Exporting files from UNIGRAPH is limited to 8 datasets of 16 characters.

b.) Final data - headers

Header descriptions:

date Julian date.decimal time (GMT).
xCO2a the measured value of xCO2 in the sea surface air sampled at the ship's bow (umol/mol).
xCO2s the measured value of xCO2 in the seawater (umol/mol).
H2Oa the measured value of H2O in the air (mmol/mol).
H2Os the measured value of H2O in the seawater (mmol/mol).
tempa the temperature of the water flowing in the equilibrator corresponding to the air xCO2 measurement (deg C).
temps the temperature of the water flowing in the equilibrator corresponding to the seawater xCO2 measurement (deg C).
airpress the ambient atmospheric pressure corresponding to the air xCO2 measurement (kPa).
seapress the ambient atmospheric pressure corresponding to the seawater xCO2 measurement (kPa).
insitemp surface temperature value (used to calculate ea). (deg C).
ea the vapor pressure (calculated using insitemp)
es the vapor pressure (calculated using temps)
pCO2air the final calculated value of pCO2a (corrected to 100% humidity,using ea and airpress) in sea surface air (uatm).
pCO2eq the final calculated value of pCO2s (corrected to 100% humidity,using es and seapress) in seawater (uatm).
lat the latitude corresponding to the ship position at the time of the pCO2 measurement.
long the longitude corresponding to the ship position at the time of the pCO2 measurement.
salts The salinity of the surface water as measured by the ship's thermo-salinometer.

c.) Final data - read.me file

This is a copy of newread.me from JGOFS_EQPAC and is a good example of an informative read.me file.

17 Nov. 1993 (updated)

This file contains information describing the pCO2 data collected during JGOFS EQPAC cruise TT007, TT011 and TT012.

The file name describes the contents.

For example:

tt007air.dat Contains the sea surface air measurements of xCO2, collected during the Spring survey leg; Cruise tt007. Includes Date,xCO2a,H2Oa,tempa,airpress,insitemp,ea and wetair.

Each file contains headers describing the columns.

Header descriptions:

date Julian date.decimal time (GMT).
xCO2a the measured value of xCO2 in the sea surface air sampled at the ship's bow (umol/mol).
xCO2s the measured value of xCO2 in the seawater (umol/mol).
H2Oa the measured value of H2O in the air (mmol/mol).
H2Os the measured value of H2O in the seawater (mmol/mol).
tempa the temperature of the water flowing in the equilibrator corresponding to the air xCO2 measurement (deg C).
temps the temperature of the water flowing in the equilibrator corresponding to the seawater xCO2 measurement (deg C).
airpress the ambient atmospheric pressure corresponding to the air xCO2 measurement (kPa).
seapress the ambient atmospheric pressure corresponding to the seawater xCO2 measurement (kPa).
insitemp surface temperature value (used to calculate ea). (deg C).
ea the vapor pressure (calculated using insitemp)
es the vapor pressure (calculated using temps)
pCO2air the final calculated value of pCO2a (corrected to 100% humidity,using ea and airpress) in sea surface air (uatm).
pCO2eq the final calculated value of pCO2s (corrected to 100% humidity,using es and seapress) in seawater (uatm).
lat the latitude corresponding to the ship position at the time of the pCO2 measurement.
long the longitude corresponding to the ship position at the time of the pCO2 measurement.
salts The salinity of the surface water as measured by the ship's thermo-salinometer.

REMARKS:

1) "?'s" represent eliminated data (due to sample contaimination by ship's exhaust).
2) "99's" in insitemp or salts column represent data collection failure by DAS system.
3) xCO_2 data has been corrected to gas standards and averaged over 4 minutes.
4) H_2O, equilibrator temperature and ambient pressure, have been averaged over 4 minutes.
5) insitemp is the corrected, estimated or interpolated sea surface water temperature.
6) Salinity, surface temperature and ship's position have been taken from the R/V Thomas G. Thompson, University of Washington, DAS files.

Appendix C

Programs

```
pro wpco2
        reftime=julday(12,31,1992)    ;changed year to 1992
        line='
        co2=fltarr(5) & h2o=co2 & temp=co2 & press=co2 & std=co2 & time=co2
        on_ioerror,finish
        openr,listunit,'pco2.lis',/GET_LUN
        infile='
        for ifile=0,36 do begin
        readf,listunit,infile
        outfile=infile+'.corr'

        openr,inunit,infile,/get_lun
        openw,outunit,outfile,/get_lun
        for iline=0,2 do readf,inunit,line                 ; skip header lines

        for isamp=0,1000 do begin                          ; begin main loop
here:
         for iline=0,4 do begin                            ; read low std info
          readf,inunit,line
          bline=byte(line)
          std(iline)=float(string(bline(20:27)))
         endfor
         lstd=avg(std(1:4))

         for iline=0,4 do begin                            ; read high std
          readf,inunit,line
          bline=byte(line)
          std(iline)=float(string(bline(20:27)))
         endfor
         hstd=avg(std(1:4))

         a=(408.72-278.69)/(hstd-lstd)                                               ; compute
coefficients
         b=408.72-(a*hstd)
;
         for imeas=0,4 do begin                            ; 5 pairs of air, eqm readings


          for iline=0,4 do begin                           ; air readings
           readf,inunit,line
           bline=byte(line)
           field=string(bline(0:1))
```

```
    month=fix(string(bline(2:3)))             ; date/time -> julian day
    day=fix(string(bline(5:6)))
tyr=string(bline(8:9))
year=fix(strcompress('19'+tyr,/remove_all))
    hour=fix(string(bline(11:12)))
    minute=fix(string(bline(14:15)))
    jd=julday(month,day,year)-reftime
    time(iline)=float(jd)+hour/24.+minute/1440.
    co2(iline)=float(string(bline(21:26)))
    h2o(iline)=float(string(bline(29:34)))
    temp(iline)=float(string(bline(49:55)))
    press(iline)=float(string(bline(56:62)))
  endfor
  meanairco2=a*avg(co2(1:4))+b
  meanairh2o=avg(h2o(1:4))
  meanairtemp=avg(temp(1:4))
  meanairpress=avg(press(1:4))
  meanairtime=avg(time(1:4))

  for iline=0,4 do begin                      ; eqm readings
    readf,inunit,line
    bline=byte(line)
    field=string(bline(0:1))
    month=fix(string(bline(2:3)))
    day=fix(string(bline(5:6)))
tyr=string(bline(8:9))
year=fix(strcompress('19'+tyr,/remove_all))
;   year=fix(string(bline(8:9)))
    hour=fix(string(bline(11:12)))
    minute=fix(string(bline(14:15)))
    jd=julday(month,day,year)-reftime
    time(iline)=float(jd)+hour/24.+minute/1440.
    co2(iline)=float(string(bline(21:26)))
    h2o(iline)=float(string(bline(29:34)))
    temp(iline)=float(string(bline(49:55)))
    press(iline)=float(string(bline(56:62)))
  endfor
  meaneqmco2=a*avg(co2(1:4))+b
  meaneqmh2o=avg(h2o(1:4))
  meaneqmtemp=avg(temp(1:4))
  meaneqmpress=avg(press(1:4))
```

```
    meaneqmtime=avg(time(1:4))

  outputtime=avg([meanairtime,meaneqmtime])

  printf,outunit,outputtime,meanairco2,meaneqmco2,meanairh2o,meaneqmh2o,$
              meanairtemp,meaneqmtemp,meanairtime,meaneqmtime,$
    meanairpress,meaneqmpress,$
              format='(11(1x,f7.3))'


   endfor                                              ; imeas=0,4
  endfor                                                           ; isamp=0,1000

finish:
  close,inunit
  free_lun,inunit
  close,outunit
  free_lun,outunit
  endfor                                                           ; end of file list

  close,listunit
  FREE_LUN,listunit
end                                                                ; end program
```

```
pro wstrip,infile

line='
openr,1,infile
openw,2,infile+'.strip'

on_ioerror,endfile

for i=0,4100 do begin                    ; =0,1 for pos files
getanother:
  readf,1,line
  bline=byte(line)
  if string(bline(37:37)) ne 'N' then goto,getanother
  line=string(bline(0:n_elements(bline)-2))
  printf,2,line
endfor

endfile:
close,1
close,2

end
```

```
pro dasplotw,dasfile   ;iflag
iflag=1
refday=julday(12,31,1992)
nmax=4091
nsmoo=5
time=fltarr(nmax) & lat=fltarr(nmax) & lon=lat
temp=lat & salt=lat & speed=lat & heading=lat
date=strarr(nmax) & gmt=date

openr,unit,dasfile, /GET_LUN
ON_IOERROR, loopend                                  ; if read bad line of data, ignore it
iline=0
for ii=0,nmax-1 do begin
   dataline='
   readf, unit, dataline                             ; read a line as a string
   dataline=byte(dataline)                                   ; convert to array
of bytes
                                                                           ;
then subarrays back to string
   date(iline)=string(dataline(0:7))
   gmt(iline)=string(dataline(9:16))

   month=fix(string(dataline(0:1))) & day=fix(string(dataline(3:4)))
   year=fix(string(dataline(6:7)))+1900 & hour=fix(string(dataline(9:10)))
   min=fix(string(dataline(12:13))) & sec=fix(string(dataline(15:16)))
   jd=julday(month,day,year)-julday(12,31,year-1)
   time(iline)=float(jd)+hour/24.+min/1440.

   lat(iline)=float(string(dataline(26:28))) + $          ; convert subarrays to real
         float(string(dataline(29:35)))/60.
   lon(iline)=float(string(dataline(39:41))) + $
         float(string(dataline(43:49)))/60.

   if string(dataline(18)) eq 'S' then lat(iline)=-lat(iline)
   if string(dataline(30)) eq 'W' then lon(iline)=-lon(iline)
   heading(iline)=float(string(dataline(69:73)))
   speed(iline)=float(string(dataline(57:60)))

   if iflag eq 1 then begin
    temp(iline)=float(string(dataline(83:88)))
    salt(iline)=float(string(dataline(97:102)))
   endif else begin
```

```
    temp(iline)=-99.
    salt(iline)=-99.
  endelse
  iline=iline+1
loopend:
endfor

endfile:
close,unit
FREE_LUN, unit
nlines=iline
print,' got ',nlines,' good data records',format='(a5,i4,a)'

time=bandav(time(0:nlines-1),nsmoo) & heading=bandav(heading(0:nlines-1),nsmoo)
speed=bandav(speed(0:nlines-1),nsmoo)
lat=bandav(lat(0:nlines-1),nsmoo)
lon=bandav(lon(0:nlines-1),nsmoo) & temp=bandav(temp(0:nlines-1),nsmoo)
salt=bandav(salt(0:nlines-1),nsmoo)
idate=0

for ii=0,nlines-nsmoo,nsmoo do begin
  date(idate)=date(ii+nsmoo/2)
  idate=idate+1
endfor

nlines=n_elements(time)
print,' band averaged to ',nlines,' records',format='(a26,i4,a)'

data=fltarr(7,nlines)
data(0,*)=time(0:nlines-1) & data(1,*)=heading(0:nlines-1) & data(2,*)=speed(0:nlines-1)
data(3,*)=lat(0:nlines-1) & data(4,*)=lon(0:nlines-1) & data(5,*)=temp(0:nlines-1)
data(6,*)=salt(0:nlines-1)

openw,1,dasfile+'.fmt'
printf,1,dasfile
printf,1,' JD1993  Heading   Speed    Lat   Lon    Temp    Salt'
for iline=0,nlines-1 do printf,1,data(*,iline),format='(7f10.4)'
close,1

!p.multi=[0,1,2,0,0]
wh=where( salt(0:nlines-1) gt 30. and salt(0:nlines-1) lt 36.5 and abs(lat) lt 17.)
```

```
if wh(0) eq -1 then goto,here

xmin=max(lat(wh))+0.15 & xmax=min(lat(wh))-0.15

spread=abs(max(salt(wh))-min(salt(wh)))
ymin=min(salt(wh))-spread/15. & ymax=max(salt(wh))+spread/15.
set_xy,xmin,xmax,ymin,ymax
plot,lat(wh),salt(wh),ytitle='Salinity (psu)',xtitle='Latitude',$
title='DAS underway data: '+dasfile,xrange=[xmin,xmax],/xstyle

wh=where( temp(0:nlines-1) gt 24.5 and temp(0:nlines-1) lt 29. )
spread=abs(max(temp(wh))-min(temp(wh)))
ymin=min(temp(wh))-spread/15. & ymax=max(temp(wh))+spread/15.
set_xy,xmin,xmax,ymin,ymax
plot,lat(wh),temp(wh),ytitle='Temp !9%!XC',xtitle='Latitude',xrange=[xmin,xmax],/xstyle

here:
end
```

```
function bandav,in,iband
;
; function BANDAV will band-average an input array,
; using npts as a window width.  Also known as
; a boxcar filter.
;
npts=n_elements(in)
nout=fix(npts/iband)
out=fltarr(npts/iband)

for ii=0,nout-1 do out(ii)=avg(in(ii*iband:(ii+1)*iband-1))

return,out
end
```

```
pro mer

line='
co2file=''
openr,dasunit,'das.dat',/GET_LUN
count=0
final=fltarr(7,123000)

while not eof(dasunit) do begin
filename='123456789123'
readf,dasunit,filename
;print,filename

;close,dasunit
;FREE_LUN,dasunit
openr, fileunit, filename,/GET_LUN
a=fstat(fileunit)
bytes=a.size
readf,fileunit,line                              ; header line
readf,fileunit,line                              ; header line
;print,line
;data=fltarr(7,8867)
no_lines=(bytes-220)/71 +2
data=fltarr(7,no_lines)
count=count+no_lines

readf,fileunit,data
close,fileunit
FREE_LUN,fileunit

final(*,count:count+no_lines-1)=data

count=count+no_lines
end

close,dasunit
FREE_LUN,dasunit

ndata=fltarr(7,count)
ndata=final(0:6,0:count-1)
;data=0
```

```
;data=ndata

time=ndata(0,*) & lat=ndata(3,*) & lon=ndata(4,*)
temp=ndata(5,*) & salt=ndata(6,*)
openr,listunit,'pco2.lis',/GET_LUN

for ifile=0,36 do begin
readf,listunit,co2file
print,'co2file',co2file
position=strpos(co2file,'.',0)
newfile=strmid(co2file,0,position)
openw,outputunit,newfile+'.mrg',/GET_LUN
openr,co2unit,newfile+'.corr',/GET_LUN
;print,co2unit,outputunit
dataline=fltarr(11,1)
outputline=fltarr(15,1)


while not eof(co2unit) do begin
;for iline=0,1000 do begin
  readf,co2unit,dataline
  outputline(0:10)=dataline
  wh=where(time ge dataline(0))
  outputline(11)=lat(wh(0)) & outputline(12)=lon(wh(0))
  outputline(13)=temp(wh(0)) & outputline(14)=salt(wh(0))
  printf,outputunit,outputline,format='(15f9.3)'
endofloop:
end
;endfor

finish:
close,co2unit
FREE_LUN,co2unit
close,outputunit
FREE_LUN,outputunit
endfor                                  ; end of list file
close,listunit
FREE_LUN,listunit
end
```

```
'program: LICOR2n.BAS
'LICOR    modifications (21dec92 dks)
'        1) eliminate code generating CO2 & H2O corrected to 1 atm
'        2) LI-COR now output CO2 (mv) and H2O (mv) which have been
'           added to the list of variables displayed and saved to disk.
' This program is identical to the one shipped to sea ~jan92
'  except that the eqn's for the pressure transducers have been
'  set for the PCO2(2) apparatus setup.
'  The ambient pressure eqn has been changed to reflect the transducer
'  in use (s/n 308420) (calib. data as of 9aug91)  24feb92 dks
'  The Li-Cor pressure eqn has been changed to reflect the transducer
'  in use (s/n 302581) (calib. data as of 30jun92)  16sept92  dks
'
'**********************************************************************
'*        Quick Basic Program for IR-CO2 Equilibrator
'*
'**********************************************************************
'
'LICOR is a modified version of IR_CO2AE which includes additional input
'       from a transducer to measure atomospheric pressure and from a
'       pCO2 value corrected to 1 atm (101.325kPa). Also the calibrate
'       routine has been modified to give averages rather than discrete
'       values for zero and span adjustment purposes. The key
'       sequence to terminate a routine has been changed from <ENTER>
'       to <CTRL ENTER>. - dks 18dec91
'IR_CO2AE includes an option for continuous standard and sample cycling
'       where high and low standards are sampled at specified time
'       intervals and air and equilibrator samples are run continuously
'       for the remainder of the cycle time.
'data handling modifications - dks 2jul91
'------------------ define variables ------------------------------
'
' DIO%() = data input or output from A/D board (DASCON-1)
' PDATA() = data storage array for temperature and pressures
' MD% = mode number for DASCON-1
' CH% = A/D channel number
' irdata$() = data storage array for pressure (Licor) corrected output
'          from Licor

DECLARE SUB DASCON1 (MD%, CH%, BYVAL DUMMY%, BYVAL DUMY%, BASADR%)
'initialize DASCON-1 drivers
DECLARE SUB diskoff ()
DECLARE SUB hexlook (s$)
DECLARE SUB line25 (f$)
DECLARE FUNCTION stopmid! (x!)

DIM dio%(8), pdata(3), eqtemp!(1024), irdata$(1024)
DIM ambbarpress!(1024), stype$(6)
COMMON SHARED dio%(), CH%, diskflag$
OPEN "com1:9600,n,8,1,asc" FOR RANDOM AS #1       'Communication port for LI-COR
BASADR% = &H300                          'DASCON-1 Data acquisition board address
avgnum% = 10                'Number of readings to average
readint% = 6               'Number of seconds between readings
stype$(1) = "LSTD": stype$(2) = "HSTD": stype$(3) = "AIR "
stype$(4) = "EQUI": stype$(5) = "0CO2": stype$(6) = "SH2O"
```

```
message$ = SPACE$(80)
diskflag$ = "false"          'Disk storage flag
'
'first switch Dascon1 to default of all ports low (to equilibrator air)
dio%(0) = 0
GOSUB setdascon1
'
'-------- Select desired sample configuration ---------------------
'
 PRINT : PRINT
 PRINT "This program assumes that the PCO2(2) apparatus is in use."
 PRINT "The apparatus configuration currently has SETRA pressure transducer"
 PRINT "   s/n 302581 to measure the LI-COR's internal pressure and"
 PRINT "   SETRA pressure transducer s/n 308420 to measure the ambient"
 PRINT "   pressure.  Either LI-COR infrared analyzer may be used."
 PRINT "   Transducer and LI-COR calibration data last updated in this"
 PRINT "   program 16sept92."
 PRINT : PRINT "Key <Enter> to continue program execution"
 DO
 LOOP WHILE INKEY$ <> CHR$(13)
 CLS : LOCATE 2, 1
DO
  keyint$ = "false"           'Keyboard interrupt flag
  onepass$ = "false"           'Flag for getreadings routine - if true,
                               ' pop out after one average
  PRINT "Program:  LICOR2     "
  PRINT : PRINT "   MAIN MENU for IR-CO2 analysis ": PRINT
  PRINT "DATE: "; DATE$; "     TIME: "; TIME$: PRINT
  PRINT "Select from the following configurations:"
  PRINT "  <0> - STOP PROGRAM"
  PRINT "  <1> - Equilibrator air"
  PRINT "  <2> - Sample air"
  PRINT "  <3> - Hi ppm CO2 standard"
  PRINT "  <4> - Low ppm CO2 standard"
  PRINT "  <5> - Calibrate"
  PRINT "  <6> - Turn disk storage on"
  PRINT "  <7> - Turn disk storage off"
  PRINT "  <8> - Adjust data input time interval"
  PRINT "        Currently "; readint%; " seconds between readings"
  PRINT "        with "; avgnum%; " readings averaged."
  PRINT "  <9> - Sample analysis with periodic standard gas measurements"
  PRINT "  <A> - Apparatus setup"
  IF diskflag$ = "false" THEN
    message$ = "No disk storage                    "
    messlen = LEN(message$)
    message$ = message$ + SPACE$(80 - messlen)
    CALL line25(message$)
  END IF
  INPUT "Desired configuration number <0-9 or A>? ", a$
   IF a$ = "0" THEN
        EXIT DO
   ELSEIF a$ = "1" THEN        'sample analysis
        dio%(0) = 0      'I/O PB0-7 all low
        itype% = 4
        GOSUB setdascon1
        PRINT "type <Ctrl Enter> to exit": PRINT
```

```
        GOSUB getreadings
    ELSEIF a$ = "2" THEN   'air analysis
        dio%(0) = 130     'I/O PB1,7 high
        itype% = 3
        GOSUB setdascon1
        PRINT "type <Ctrl Enter> to exit": PRINT
        GOSUB getreadings
    ELSEIF a$ = "3" THEN   'high standard
        dio%(0) = 138     'I/O PB1,3,7 high
        itype% = 2
        GOSUB setdascon1
        PRINT "type <enter> to exit": PRINT
        GOSUB getreadings
    ELSEIF a$ = "4" THEN   'low standard
        dio%(0) = 170     'I/O PB1,3,5,7 high
        itype% = 1
        GOSUB setdascon1
        PRINT "type <enter> to exit": PRINT
        GOSUB getreadings
    ELSEIF a$ = "5" THEN   'calibrate
        GOSUB calibrate
    ELSEIF a$ = "6" THEN   'disk storage option chosen
        GOSUB diskon
    ELSEIF a$ = "7" THEN   'disk storage off
        CALL diskoff
    ELSEIF a$ = "8" THEN   'adjust data input time interval
        GOSUB timeint
    ELSEIF a$ = "9" THEN   'sample & std. gas measurements
        GOSUB sampstd
    ELSEIF UCASE$(a$) = "A" THEN
        GOSUB apparatus
    ELSE
        PRINT "["; a$; "] is not a valid entry. Please re-enter"
  END IF
LOOP
dio%(0) = 0              'set valves to equilibrator
GOSUB setdascon1
CLOSE #1                 'communication port for LI-COR
IF diskflag$ = "true" THEN CLOSE #2    'close disk file if open
CLS
END
'*****************************************************************
'*****************************************************************

setdascon1:
dio%(1) = 0    'I/O PC0-3 all low
'
'---------------- Enter call routine ----------------------------
'
MD% = 9              'DASCON-1 mode 9 for digital I/O
CALL DASCON1(MD%, CH%, VARPTR(dio%(0)), dio%(1), BASADR%)
   GOSUB DERROR
RETURN
'*****************************************************************

getreadings:
'--------------- Read data from LI-COR ----------------------------
'
```

```
'CLS : LOCATE 2, 1
CALL line25(message$)
keyint$ = "false"
PRINT #1, "*1322,32,42,43,21,31"     'sets print list for header and data
        'list contains CO2(umol/mol), H2O(mmol/mol), licor temp.(deg C),
        '    licor bar. pressure(kPa), CO2(mv), H2O(mv)
PRINT #1, "*140"    'turn auto print off
PRINT #1, "*150"    'turn auto header off

z% = 0                        'data counter
start! = TIMER                   'start timer
DO

        z% = z% + 1                 'increment data counter
        GOSUB presstemp                'get bar. pressure and equil. temp
        'output box pressure to LI-COR
        z$ = STR$(barpress!)
        x$ = "*77" + LTRIM$(z$)  'strip off leading blank & add to press code
        PRINT #1, x$          'output to LI-COR
        IF z% = 1 THEN
         PRINT "Current readings:"
         PRINT #1, "*11"           'asks for header
         INPUT #1, w$
         PRINT TAB(11); "        CO2    H2O   Licor  Licor Equil  amb   CO2     H2O"
         PRINT TAB(11); " time   pcorr  pcorr  temp  press temp  press "
         PRINT TAB(11); "        um/m   mm/m  deg C  kPa  deg C  kPa    mv     mv "
         format$ = "\ \     & ####.## ###.### ##.### ###.## ##.## ###.## #####.# #####.##"
        END IF
        beginwait! = TIMER
        DO      'wait 2.5 sec before bringing in LICOR press corr CO2
         IF INKEY$ = CHR$(10) THEN
          keyint$ = "true"
         END IF
         endwait! = stopmid!(beginwait!)
        LOOP WHILE endwait! - beginwait! < 2.5 AND keyint$ = "false"
        PRINT #1, "*12"            'asks for data
        INPUT #1, irdata$(z%)     'saves press corr. data string in array

        IF keyint$ = "true" THEN EXIT DO  'leave getreadings main loop

        co2 = VAL(MID$(irdata$(z%), 1, 7))
        h2o = VAL(MID$(irdata$(z%), 9, 7))
        temp = VAL(MID$(irdata$(z%), 18, 7))
        press = VAL(MID$(irdata$(z%), 27, 7))
        co2mv = VAL(MID$(irdata$(z%), 36, 7))
        h2omv = VAL(MID$(irdata$(z%), 45, 7))

        PRINT USING format$; stype$(itype%); TIME$; co2; h2o; temp; press; eqtemp!(z%);
ambbarpress!(z%); co2mv; h2omv
        currentline% = CSRLIN
        LOCATE currentline% - 1
    'CALL hexlook(irdata$(z%))       'look at hex version of data if
                                      ' needed
        DO                      'this loop waits until readint%
                                      ' secs have elapsed
         stopt! = stopmid!(start!)
```

```
            IF INKEY$ = CHR$(10) THEN
              keyint$ = "true"
            END IF
          LOOP WHILE stopt! - start! < readint% AND keyint$ = "false"
          start! = stopt!                  'reset timer start

          IF z% = avgnum% THEN             'if avgnum% readings have been
            GOSUB average                  ' taken calc averages (and
                                            ' store on disk)
            z% = 0                 'reset counter
            IF onepass$ = "true" THEN RETURN   'calling routine was SAMPSTD
                                            ' so only do one average then
                                            ' exit
          END IF

LOOP WHILE INKEY$ <> CHR$(10) AND keyint$ = "false"
RETURN

'*************************************************************************

presstemp:
'
'---------------- Enter call routine ----------------------------
'
MD% = 0              'DASCON-1 mode 0 for data acquisition
' Ch. 0 = DIO%(0) = output from equilibrator thermistor
' Ch. 1 = DIO%(1) = output from baro. pressure sensor (just outside LI-COR)
' Ch. 2 = dio%(2) = output from barometric pressure sensor (ambient)
 CALL DASCON1(MD%, CH%, VARPTR(dio%(0)), dio%(1), BASADR%)
'
'---------------- Read out data -------------------------------
'
FOR i% = 0 TO 2                    'Ch. 0, Ch. 1 and Ch. 2
pdata(i%) = dio%(i%) * 2.0475 / 4096! 'convert digital data to voltage
NEXT i%                            '-4096=-2.0475 volts, 4096=2.0475 volts
'
'---------------- Calculate pressure and temperature ----------------
'
'see data sheets in LI-COR manual for conversion values
'Note that Setra transducer calibration data are based on 0-5v range and
' that our circuits are in the 0-2v range. Our voltages will be multiplied
' by a factor of 2.5 to adjust for the different voltage ranges
barpress! = ((pdata(1) * 2.5) * 100.0256 + 599.9171) / 10!   'LI-COR internal pressure in kPa
(s/n 302581 calib 30jun92)
ambbarpress!(z%) = ((pdata(2) * 2.5) * 99.99693 + 599.9035) / 10! 'Ambient barometric
pressure in kPa (s/n 308420 calib 9aug91)
'For thermistor resistance the following equation is based on 5volts
resist! = 100 / (pdata(0) + 2.5) - 20!    'Thermistor resistance
eqtemp!(z%) = 1! / (.002726 + .0006207 * LOG(resist!) / LOG(10!) + 7.091E-06 * (LOG(resist!) /
LOG(10!)) ^ 3) - 273.15  'Equilibrator temperature in celsius
'
RETURN
'
'*************************************************************************
'---------------- Calibrate LI-COR --------------------------------
'
calibrate:
```

```
DO     'start entire calibration routine loop
 DO    'start CO2 calibration routine loop
'  ******zero*********
  dio%(0) = 0    'I/O PB0-7 all low
  GOSUB setdascon1
  itype% = 5
  CLS : LOCATE 2, 1
  PRINT "          CALIBRATE"
  PRINT "The reference and sample paths should be CO2 and H2O free"
  PRINT "Adjust ZEROS for CO2 and H2O free gas when signal has stabilized"
  PRINT "and then key <CTRL ENTER>"
  PRINT : PRINT
  GOSUB getreadings
'  ******CO2 span**********
  dio%(0) = 138      'Open valve to high ppm standard (I/O PB1, 3, 7 high)
  GOSUB setdascon1
  itype% = 2
  CLS : LOCATE 2, 1
  PRINT "          CALIBRATE"
  PRINT "The sample path now contains the HIGH CO2 standard"
  PRINT "Adjust span to read reference gas (check label) when signal"
  PRINT "  has stabilized and then key <CTRL ENTER>"
  PRINT : PRINT
  GOSUB getreadings
'  *****Check low standard*************
  dio%(0) = 170      'Open valve to low ppm standard (I/O PB1,3,5,7 high)
  itype% = 1
  GOSUB setdascon1
  CLS : LOCATE 2, 1
  PRINT "       CHECK CALIBRATION": PRINT
  PRINT "The sample path now contains the LOW CO2 standard"
  PRINT "When signal has stabilized check linearity by comparing"
  PRINT "  signal with reference gas (check label).  Then key <CTRL ENTER>"
  PRINT
  GOSUB getreadings
  PRINT : PRINT
  DO
   PRINT "1  Redo CO2 calibration"
   PRINT "2  Proceed with H2O span calibration"
   INPUT "Select 1 or 2 "; a$
   IF a$ = "1" OR a$ = "2" THEN
      EXIT DO
   ELSE
      PRINT "["; a$; "] is not a valid entry.  Please re-enter"
   END IF
  LOOP
  IF a$ = "2" THEN EXIT DO  'proceed with H2O span calibration
 LOOP                   'end CO2 calibration loop
' Start H2O Calibration loop
 dio%(0) = 0             'I/O PB0-7 all low
 GOSUB setdascon1
 itype% = 6
 CLS : LOCATE 2, 1
 PRINT "          CALIBRATE": PRINT
 PRINT "Set dew point generator and valves so that the H2O span can"
 PRINT "  be adjusted when the signal has stabilized.  Then key"
```

```
 PRINT "  <CTRL ENTER>"
 PRINT : PRINT
 GOSUB getreadings
 PRINT : PRINT
 DO
  PRINT "1  Redo entire calibration"
  PRINT "2  Exit calibration routine"
  INPUT "Select 1 or 2 "; a$
  IF a$ = "1" OR a$ = "2" THEN
   EXIT DO
  ELSE
   PRINT "["; a$; "] is not a valid entry.  Please re-enter."
  END IF
 LOOP
 IF a$ = "2" THEN EXIT DO
LOOP              'end entire calibration routine loop
PRINT : PRINT
PRINT "Reset valves to default (equilibrator) setting."
RETURN
'
'***********************************************************************
'-------------- DASCON-1 error check ------------------------------
'
DERROR:

IF dio%(8) <> 0 THEN PRINT : PRINT " !!!!!!!! ERROR !!!!!!!": GOTO 500
RETURN

500  PRINT : PRINT "ERROR FLAGS": PRINT "----------": PRINT
 PRINT "Mode or channel error DIO%(8) = "; HEX$(dio%(8))
 PRINT : PRINT : PRINT "Error codes (hex):-"
 PRINT "                0 - Normal, no error"
 PRINT "               10 - Mode number MD% out of range"
 PRINT "              300 - A/D time out"
 PRINT "              500 - Overload i.e. input beyond +/- FS"
 PRINT "              600 - Configuration error, CH% out of range"
 PRINT "              700 - Data range error, DIO%(0) <0 OR >255"
 PRINT "                         or DIO%(1) <0 or >15"
 PRINT "              800 - A/D & Digital mode error i.e. selecting"
 PRINT "                    A/D modes 1,2 or 3 with digital config. not 0"
990 STOP
'***********************************************************************
diskon:          'turn on disk storage
IF diskflag$ = "true" THEN      'disk file already open
 PRINT : PRINT "Disk file "; filename$; " is already open."
 PRINT "It must be closed before a new one can be opened."
 DO
  PRINT "1  Close "; filename$; " now"
  PRINT "2  Exit this routine to main menu "
  INPUT "Select 1 or 2 "; a$
  IF a$ = "1" THEN     'close file already open
   CALL diskoff
   diskflag$ = "false"
   EXIT DO
  ELSEIF a$ = "2" THEN
   RETURN              'return to main menu
```

```
    ELSE
      PRINT "["; a$; "] is not a valid entry.  Please re-enter"
    END IF
  LOOP
END IF
namemodeflag$ = "false"
ON ERROR GOTO errsort

numsecs! = readint% * avgnum%
PRINT "Data readings will be averaged and stored every "; numsecs!; " secs"
DO                'remain in this loop until namemodeflag$ = "true"
  PRINT
  errflag$ = "false"   'set false before each try on OPEN __ for INPUT
  INPUT "Enter path and filename for data storage "; filename$
  OPEN filename$ FOR INPUT AS #2

  IF errflag$ = "false" THEN    'file already exists
    PRINT "File "; filename$; " already exists!"
    PRINT "  <1>  change filename"
    PRINT "  <2>  append new data to this file"
    PRINT "  <3>  overwrite all data in this file"
    DO
      INPUT "Select option (1, 2 or 3) "; op$
    LOOP WHILE INSTR("123", op$) = 0
    CLOSE #2                 'just opened #2 for checking
    'op$ = "1"  will merely cause another loop through filename input

    IF op$ = "2" THEN          'append option chosen
      OPEN filename$ FOR APPEND AS #2
      namemodeflag$ = "true"
      message$ = "Data will be appended to file " + filename$
      messlen = LEN(message$)
      message$ = message$ + SPACE$(80 - messlen)
    ELSEIF op$ = "3" THEN       'overwrite option chosen
      PRINT "Overwrite file "; filename$; " option chosen"
      DO
          INPUT "Are you sure (y/n) "; ans$
      LOOP WHILE INSTR("YyNn", ans$) = 0
      IF LCASE$(ans$) = "y" THEN
          OPEN filename$ FOR OUTPUT AS #2
          PRINT #2, TAB(11); "          CO2    H2O   Licor  Licor Equil   amb    CO2    H2O"
          PRINT #2, TAB(11); "  time    pcorr  pcorr  temp   press temp   press  "
          PRINT #2, TAB(11); "          um/m   mm/m   deg C  kPa  deg C   kPa    mv     mv"
          namemodeflag$ = "true"
          message$ = "Starting at " + TIME$ + " data is overwriting contents of " + filename$
          messlen = LEN(message$)
          message$ = message$ + SPACE$(80 - messlen)
      ELSE
          PRINT "Re-enter path and filename"
      END IF          'end overwrite if
    END IF            'end filename exists if

  ELSEIF errflag$ = "true" AND errnum% = 53 THEN    'file not found
    OPEN filename$ FOR OUTPUT AS #2           'create new file
    PRINT #2, TAB(11); "          CO2    H2O   Licor  Licor Equil   amb    CO2    H2O"
    PRINT #2, TAB(11); "  time    pcorr  pcorr  temp   press temp   press  "
```

```
  PRINT #2, TAB(11); "        um/m   mm/m   deg C   kPa  deg C   kPa     mv     mv"
  namemodeflag$ = "true"
  message$ = "file " + filename$ + " created"
  messlen = LEN(message$)
  message$ = message$ + SPACE$(80 - messlen)

 END IF

LOOP WHILE namemodeflag$ = "false"     'go back to re-enter filename

CALL line25(message$)
ON ERROR GOTO 0                  'turn off error trapping
diskflag$ = "true"
RETURN

'************************************************************************

errsort:
errnum% = ERR
IF errnum% = 76 THEN
 PRINT : PRINT USING "Pathname portion of & is inaccessible "; filename$
ELSEIF errnum% <> 53 AND errnum% <> 76 THEN
 'err = 53 => file not found (new one will be created)
 PRINT "Other error:  "; errnum%; "   Program execution continues..."
END IF
errflag$ = "true"
RESUME NEXT
'************************************************************************
timeint:        'set readings time interval and number of readings for avg
PRINT "Readings will be taken every "; readint%; " secs"
INPUT "enter new readings time interval (not less than 4 sec) "; readsec%
IF readsec% >= 4 THEN readint% = readsec%
PRINT "data will be averaged over "; avgnum%; " readings"
INPUT "enter new number of readings over which to average "; anum%
IF anum% > 0 THEN avgnum% = anum%
PRINT
PRINT "Readings will be taken every "; readint%; " secs"
PRINT "  and averaged over "; avgnum%; " readings"
PRINT "  which is equivalent to an average every "; readint% * avgnum%; " secs"
PRINT "  or an average every "; readint% * avgnum% / 60; " min."
PRINT : PRINT
RETURN
'************************************************************************
average:       'average avgnum% readings (and save on disk if option chosen)
dateval$ = DATE$
timeval$ = TIME$
seqt = 0: co2 = 0: h2o = 0: bt = 0: bp = 0: ambp = 0: co2mv = 0: h2omv = 0
FOR j = 1 TO z%
 seqt = seqt + eqtemp!(j)
 co2 = co2 + VAL(MID$(irdata$(j), 1, 7))
 h2o = h2o + VAL(MID$(irdata$(j), 9, 7))
 bt = bt + VAL(MID$(irdata$(j), 18, 7))
 bp = bp + VAL(MID$(irdata$(j), 27, 7))
 ambp = ambp + ambbarpress!(j)
 co2mv = co2mv + VAL(MID$(irdata$(j), 36, 7))
 h2omv = h2omv + VAL(MID$(irdata$(j), 45, 7))
NEXT j
```

```
seqtf = seqt / z%
co2f = co2 / z%
h2of = h2o / z%
btf = bt / z%
bpf = bp / z%
ambpf = ambp / z%
co2mvf = co2mv / z%
h2omvf = m2omv / z%

sampletype$ = MID$(stype$(itype%), 1, 1)
dval$ = MID$(dateval$, 1, 6) + MID$(dateval$, 9, 2)
formatf$ = "& & & ####.## ###.### ##.### ###.## ##.## ###.## #####.# #####.##"

PRINT
PRINT
PRINT "Averages over "; avgnum% * readint%; " secs"
PRINT USING formatf$; sampletype$; dval$; timeval$; co2f; h2of; btf; bpf; seqtf; ambpf;
co2mvf; h2omvf
PRINT
IF diskflag$ = "true" THEN
  PRINT #2, USING format$; sampletype$; dval$; timeval$; co2f; h2of; btf; bpf; seqtf; ambpf;
co2mvf; h2omvf
  CLOSE #2          'close to dump above contents to disk
  OPEN filename$ FOR APPEND AS #2
END IF
RETURN
**************************************************************************

sampstd:  'sample analysis with periodic standard gas measurements
onepass$ = "true"     'set flag for only one pass through getreadings routine
keyint$ = "false"     'interrupt flag
cycletime% = 7200     'total stds & measurements cycle time (two hours)
meastime% = 300       'measure each std. for 300 sec (5 min.)
stdflushtime% = 300   'flush with each std for 300 sec (5 min.) before meas.
sampflushtime% = 300  'flush with sample for 300 sec (5 min.) before meas.
saveavgnum% = avgnum% 'save preset number of readings to average
savereadint% = readint% 'save preset # of sec between samp data readings
stdreadint% = 6       '# of secs between std data readings (from LI-COR)
stdavgnum% = 10       '# of std readings
sampreadint% = readint% '# of secs between sample data readings
sampavgnum% = avgnum%   '# of sample readings to average

DO
  PRINT " <0> - Exit to Main Configuration Menu"
  PRINT " <1> - A sequence of Standard gas measurements will be started"
  PRINT "       once every "; cycletime%; " secs (once every ";
  PRINT cycletime% / 3600; " hour(s))."
  PRINT "       (ie. The low and high stds. will be sampled for "; meastime% / 60; " min(s) each"
  PRINT "       and then for the remainder of the "; cycletime% / 3600; " hour(s) the equil."
  PRINT "       and air will be sampled alternately.)"
  PRINT " <2> - Each STANDARD or SAMPLE will be measured for a period of "
  PRINT "       "; meastime%; " secs (or "; meastime% / 60; " min(s)) before "
  PRINT "       switching to the next one in the sequence."
  PRINT " <3> - Each STANDARD will be flushed through lines for "; stdflushtime%
  PRINT "       secs before being measured."
  PRINT " <4> - SAMPLE will be flushed through lines for "; sampflushtime%;
  PRINT " secs before "
```

```
PRINT "      being measured"
PRINT " <5> - Data readings will be taken from LI-COR for STANDARDS"
PRINT "      every "; stdreadint%; " secs"
PRINT " <6> - Number of STANDARDS readings to average "; stdavgnum%
PRINT " <7> - SAMPLE data readings will be taken every "; sampreadint%; " secs"
PRINT " <8> - Number of SAMPLE readings to average "; sampavgnum%
PRINT " <9> - Accept above values and CONTINUE WITH ANALYSIS"
PRINT " The above yields a configuration where a complete STANDARDS and"
PRINT " SAMPLE analysis cycle will take about "; cycletime% / 3600!; " hour(s) with"
PRINT " total STANDARD flush and analysis time (for both high and low) = ";
PRINT 2 * (meastime% + stdflushtime%) / 60; " min."
PRINT " This leaves about "; (cycletime% - 2 * (meastime% + stdflushtime%)) / 60;
PRINT " min. for sample (equil and air) analysis."
DO
  INPUT "Select a number from 0 to 9 "; iopt$
LOOP WHILE INSTR("0123456789", iopt$) = 0
IF iopt$ = "0" THEN     'exit to main configuration menu
  CLS
  RETURN
ELSEIF iopt$ = "1" THEN
  INPUT "Enter new value (secs) for total SAMPLE/STD cycletime "; newval%
  IF newval% > 0 THEN cycletime% = newval%
ELSEIF iopt$ = "2" THEN
  INPUT "Enter new time (sec) for each STANDARD and SAMPLE to be measured "; newval%
  IF newval% >= 6 THEN meastime% = newval%
ELSEIF iopt$ = "3" THEN
  INPUT "Enter new time (sec) for STANDARD to be flushed through lines "; newval%
  IF newval% > 0 THEN stdflushtime% = newval%
ELSEIF iopt$ = "4" THEN
  INPUT "Enter new time (sec) for SAMPLE to be flushed through lines "; newval%
  IF newval% > 0 THEN sampflushtime% = newval%
ELSEIF iopt$ = "5" THEN
  PRINT "Enter new time (secs) (not less than 6) between data readings from"
  INPUT "  LI-COR for STANDARDS "; newval%
  IF newval% >= 6 THEN stdreadint% = newval%
ELSEIF iopt$ = "6" THEN
  INPUT "Enter new number of STANDARDS readings to average "; newval%
  IF newval% > 0 THEN stdavgnum% = newval%
ELSEIF iopt$ = "7" THEN
  PRINT "Enter new time (secs) (not less than 6) between data readings from"
  INPUT "  LI-COR for SAMPLE "; newval%
  IF newval% >= 6 THEN sampreadint% = newval%
ELSEIF iopt$ = "8" THEN
  INPUT "Enter new number of SAMPLE readings to average "; newval%
  IF newval% > 0 THEN sampavgnum% = newval%
END IF
LOOP WHILE iopt$ <> "9"
'
PRINT : PRINT
PRINT "type <Ctrl Enter> to exit": PRINT
' now start cycle
DO                        'start cycle loop
 startcycletime! = TIMER       'start cycle timer
 avgnum% = stdavgnum%
 readint% = stdreadint%
' cycle through standards
```

```
FOR itype% = 1 TO 2
 SELECT CASE itype%
  CASE 1
     dio%(0) = 170           'choose low std
  CASE 2
     dio%(0) = 138           'choose high std
 END SELECT
 GOSUB setdascon1
 startflush! = TIMER         'start flush timer
 PRINT stype$(itype%); " flush now in progress"
 DO
  IF INKEY$ = CHR$(10) THEN keyint$ = "true"
 LOOP WHILE stopmid!(startflush!) - startflush! < stdflushtime% AND keyint$ = "false"
 IF keyint$ <> "false" THEN EXIT FOR
 startmeas! = TIMER          'start measurement timer
 DO
  GOSUB getreadings
 LOOP WHILE stopmid!(startmeas!) - startmeas! < meastime% AND keyint$ = "false"
 IF keyint$ <> "false" THEN EXIT FOR
NEXT itype%
IF keyint$ <> "false" THEN EXIT DO  'exit cycling and prepare to leave routine
'
'     now start sample cycling
avgnum% = sampavgnum%
readint% = sampreadint%
DO                           'start air and equil sample cycling
 FOR itype% = 3 TO 4
  SELECT CASE itype%
     CASE 3
      dio%(0) = 130          'choose air sample
     CASE 4
      dio%(0) = 0            'choose equil sample
  END SELECT
  GOSUB setdascon1
  startflush! = TIMER        'start sample flush timer
  PRINT stype$(itype%); " flush now in progress"
  DO
     IF INKEY$ = CHR$(10) THEN keyint$ = "true"
  LOOP WHILE stopmid!(startflush!) - startflush! < sampflushtime% AND keyint$ = "false"
  IF keyint$ <> "false" THEN EXIT FOR
  startmeas! = TIMER
  DO
     GOSUB getreadings
  LOOP WHILE stopmid!(startmeas!) - startmeas! < meastime% AND keyint$ = "false"
  IF keyint$ <> "false" THEN EXIT FOR
 NEXT itype%
'end AIR and EQUIL sample loop
LOOP WHILE stopmid!(startcycletime!) - startcycletime! < cycletime% AND keyint$ = "false"
'
LOOP WHILE keyint$ = "false"    'end cycling loop
'
avgnum% = saveavgnum%
readint% = savereadint%
dio%(0) = 0
GOSUB setdascon1
RETURN
```

```
*************************************************************************
apparatus:      'menu and routine for setting and checking appartus
CLS
DO
 PRINT : PRINT "      APPARATUS MENU"
 PRINT : PRINT "Select from the following options"
 PRINT " <0> - Return to MAIN MENU"
 PRINT " <1> - Send manufacturer's calibration data for CO2 and H2O"
 PRINT "       to LI-COR"
 PRINT " <2> - Check pressure transducer and temperature sensor readings"
 PRINT : INPUT "Enter option number <0 - 2> "; op$
 IF op$ = "0" THEN       'exit loop and return to previous menu
  EXIT DO
 ELSEIF op$ = "1" THEN   'send manufacturer's calibration data to LI-COR
  GOSUB sendlicorcalib
 ELSEIF op$ = "2" THEN   'check pressure and temp readings
  z% = 1
  GOSUB presstemp
  PRINT : PRINT
  PRINT USING "equil temp voltage (pdata(0)) = ###.### "; pdata(0)
  PRINT USING "Equilibrator temp. (deg C) = ##.### "; eqtemp!(z%)
  PRINT USING "LI-COR barom press voltage (pdata(1)) = ###.### "; pdata(1)
  PRINT USING "LI-COR barometric pressure (kPa) = ####.##"; barpress!
  PRINT USING "Ambient barom press voltage (pdata(2)) = ###.### "; pdata(2)
  PRINT USING "Ambient barometric pressure (kPa) = ####.##"; ambbarpress!(z%)
 ELSE
  PRINT "["; a$; "] is not a valid entry.  Please re-enter."
 END IF
LOOP
CLS
CALL line25(message$)
RETURN
*************************************************************************
sendlicorcalib:   'send manufacturer's calibration data to LI-COR
PRINT : PRINT
PRINT "In general, sending the calibration data is not necessary unless"
PRINT "  there has been a LI-COR hardware failure (eg. an EEPROM failure)."
DO
 PRINT " <0> - Return to APPARATUS MENU"
 PRINT " <1> - Send LI-COR #1 (s/n IRG3-150) CO2 & H2O calibration data"
 PRINT " <2> - Send LI-COR #2 (s/n IRG3-209) CO2 & H2O calibration data"
 PRINT : INPUT "Enter option number <0-2> "; cop$
 IF cop$ = "0" THEN  'return to apparatus menu
  EXIT DO
 ELSEIF cop$ = "1" THEN   'send LI-COR 1 (s/n IRG3-150) its calib. data
  'send CO2 calibration data - last calib. date 12may92
  PRINT #1, "*0135.9,19270,.1428,1.699e-05,2.352e-09,.1370,1.995e-05,2.081e-09,0,2,H"
  'send H2O calibration data
  PRINT #1, "*0240.8,15780,6.450e-03,2.948e-06,-1.868e-11,0"
  PRINT : PRINT : PRINT "CO2 and H2O calibration data sent to LI-COR #1"
  PRINT
 ELSEIF cop$ = "2" THEN   'send LI-COR 2 (s/n IRG3-209) its calib. data
  'send CO2 calibration data - last calib. date 20jul92
  PRINT #1, "*0135.4,18980,.1414,1.751e-05,2.421e-09,.1365,2.006e-05,2.176e-09,0,2,H"
  'send H2O calibration data
  PRINT #1, "*0240.3,15100,6.378e-03,3.029e-06,-4.284e-12,0"
```

```
  PRINT : PRINT : PRINT "CO2 and H2O calibration data sent to LI-COR #2"
 ELSE
  PRINT "["; cop$; "] is not a valid entry.  Please re-enter."
 END IF
LOOP
RETURN
'********************************************************************
END
'********************************************************************

'********************************************************************
SUB diskoff          'turn disk storage off
diskflag$ = "false"
message$ = "disk storage off                 "
messlen = LEN(message$)
message$ = message$ + SPACE$(80 - messlen)
CALL line25(message$)
CLOSE #2             'close diskfile
END SUB

'********************************************************************
SUB hexlook (s$)        'look at hex form of a string
PRINT
FOR i = 1 TO LEN(s$)
 PRINT HEX$(ASC(MID$(s$, i, 1))); " ";
NEXT i
PRINT
END SUB

'********************************************************************
SUB line25 (f$)         'print text on line 25
currentline% = CSRLIN
LOCATE 25: PRINT f$      'print message on line 25
LOCATE currentline%
END SUB

'********************************************************************
FUNCTION stopmid! (startval!)       'check for midnight event
 checkval! = TIMER
 IF checkval! < startval! THEN      ' if midnight passed, make adjustments
  stopmid! = checkval! + 86400
 ELSE
  stopmid! = checkval!
 END IF
END FUNCTION
```

DOCUMENT LIBRARY

Distribution List for Technical Report Exchange – July 1, 1993

University of California, San Diego
SIO Library 0175C (TRC)
9500 Gilman Drive
La Jolla, CA 92093-0175

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

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
Stennis Space Center
NSTL, MS 39522-5001

Library
Institute of Ocean Sciences
P.O. Box 6000
Sidney, B.C. V8L 4B2
CANADA

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

IFREMER
Centre de Brest
Service Documentation - Publications
BP 70 29280 PLOUZANE
FRANCE

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-94-01	2.	3. Recipient's Accession No.
4. Title and Subtitle Measuring xCO_2 Using the CAT/NDIR Method—System set-up, calibration, maintenance and shutdown			5. Report Date February 1994
			6.
7. Author(s) Maren E. Tracy, Edward T. Peltzer and Catherine Goyet			8. Performing Organization Rept. No. WHOI-94-01
9. Performing Organization Name and Address Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543			10. Project/Task/Work Unit No.
			11. Contract(C) or Grant(G) No. (C) FG02 94ER61544 (G)
12. Sponsoring Organization Name and Address Woods Hole Oceanographic Institution			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-94-01.

16. Abstract (Limit: 200 words)

Accurate measurement of partial pressure of CO_2 in seawater is currently performed by measuring pCO_2 in an aliquot of a small volume of gas equilibrated with a large volume of the seawater to be measured. PCO_2 in the gas phase can be accurately measured either by gas chromatography or infra-red analysis. In order to minimize human labor to monitor pCO_2 in surface seawater we opted for the infra-red analysis which does not require a highly trained person and which can easily be automated. This report describes how we have designed and automated a system for continual surface seawater pCO_2 monitoring. It further indicates the necessary steps to set up, run, and maintain the system. With minor modifications this system can also be used to measure pCO_2 in discrete seawater samples. (Goyet et al., 1993)

Reference:

Goyet, C., Millero F.J., Poisson A., and Shafer D.K. (1993). Temperature dependence of CO_2 fugacity in seawater. Marine Chemistry, 44, 205, 219.

17. Document Analysis a. Descriptors

CO_2 partial pressure measurement
infra-red CO_2 analysis
seawater pCO_2

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for public release; distribution unlimited.	UNCLASSIFIED	54
	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