Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

NOGAP B.6, Oxygen Isotope Data from Water and Ice Cores from the Beaufort Sea, September 1990 and May 1991

D. W. Paton, A. Abehennah, W. Grieve and R. W. Macdonald

Institute of Ocean Sciences
Department of Fisheries and Oceans
Sidney, British Columbia
V8L 4B2

1994

Canadian Data Report of Hydrography and Ocean Sciences 134

Fisheries and Oceans   Pêches et Océans

Canada

Canadian Data Report Of Hydrography and Ocean Sciences

Data reports provide a medium for the documentation and dissemination of data in a form directly useable by the scientific and engineering communities. Generally, the reports contain raw and/or analyzed data but will not contain interpretations of the data. Such compilations commonly will have been prepared in support of work related to the programs and interests of the Ocean Science and Surveys (OSS) sector of the Department of Fisheries and Oceans.

Data reports are not intended for general distribution and the contents must not be referred to in other publications without prior written authorization from the issuing establishment. The correct citation appears above the abstract of each report. Data reports are abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Data reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out of stock reports will be supplied for a fee by commercial agents.

Regional and headquarters establishments of Ocean Science and Surveys ceased publication of their various report series as of December 1981. A complete listing of these publications is published in the *Canadian Journal of Fisheries and Aquatic Sciences*, Volume 39: Index to Publications 1982. The current series, which begins with report number 1, was initiated in January 1982.

Rapport statistique canadien sur l'hydrographie et les sciences océaniques

Les rapports statistiques servent de véhicule pour la compilation et la diffusion des données sous une forme directement utilisable par les scientifiques et les techniciens. En général, les rapports contiennent des données brutes ou analysées, mais ne fournissent pas d'interprétation des données. Ces compilations sont préparées le plus souvent à l'appui de travaux liés aux programmes et intérêts du service des Sciences et levés océaniques (SLO) du ministère des Pêches et des Océans.

Les rapports statistiques ne sont pas destinés à une vaste distribution et leur contenu ne doit pas être mentionné dans une publication sans une autorisation écrite préalable de l'établissement auteur. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports statistiques sont résumés dans la revue *Résumés des sciences halieutiques et aquatiques*, et ils sont classés dans l'index annuel des publications scientifiques et techniques du Ministère.

Les rapports statistiques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés sont fournis contre rétribution par des agents commerciaux.

Les établissements des Sciences et levés océaniques dans les régions et à l'administration centrale ont cessé de publier leurs diverses séries de rapports en décembre 1981. Une liste complète de ces publications figure dans le volume 39, Index des publications 1982, du *Journal canadien des sciences halieutiques et aquatiques*. La série actuelle a commencé avec la publication du rapport numéro 1 en janvier 1982.

Canadian Data Report of Hydrography and Ocean Sciences No. 134

1994

NOGAP B.6, OXYGEN ISOTOPE DATA FROM WATER AND ICE CORES FROM THE BEAUFORT SEA, SEPTEMBER 1990 AND MAY 1991

by

D.W. Paton, A. Abehennah, W. Grieve, R.W. Macdonald

Institute of Ocean Sciences
Department of Fisheries and Oceans
Sidney, B.C.

Cat. No. Fs 97 – 16/134 ISSN 0711-6721

Correct citation for this publication:

Paton, D.W., A. Abehennah, W. Grieve and R.W. Macdonald, 1994, NOGAP B.6, Oxygen isotope data from water and ice cores from the Beaufort Sea, September 1990, and May 1991 *Can. Data Rep. Hydrogr. Ocean Sci.:* **134**, 118 pp.

Contents

Abstract

Paton, D.W., A. Abehennah, W. Grieve and R.W. Macdonald, 1994, NOGAP B.6, Oxygen isotope data from water and ice cores from the Beaufort Sea, September 1990 and May 1991 *Can. Data Rep. Hydrogr. Ocean Sci.:* **134**, 118 pp.

As part of the NOGAP B.6 program (Beaufort Sea Oceanography), with objectives to determine hydrocarbon pathways and primary productivity of the waters overlying the Mackenzie Shelf, we conducted a ship-based sampling program in August-September, 1990 (CCGS *Henry Larsen*, Insitute of Ocean Sciences I.D. #9070) and an ice-based spring sampling program (April 23-May 11, 1991, Institute of Ocean Sciences I.D. #9109). Chemical and physical measurements for these missions have been previously reported [*Macdonald et al.*, 1991, 1992]. We report here the measurements made of oxygen isotope composition ($\delta^{18}O$) and salinity for water samples collected in September, 1990 (mission #9070), and for ice-core and water samples collected in April-May, 1991 (Mission #9109).

Key words: Arctic, $\delta^{18}O$, coastal zone, ice.

Résumé

Paton, D.W., A. Abehennah, W. Grieve and R.W. Macdonald, 1994, NOGAP B.6, Oxygen isotope data from water and ice cores from the Beaufort Sea, September 1990 and May 1991 *Can. Data Rep. Hydrogr. Ocean Sci.:* **134**, 118 pp.

Dans le cadre du programme NOGAP B.6 (Océanographie de la mer de Beaufort), dont les objectifs sont de déterminer le cheminement des hydrocarbures et la productivité primaire dans les eaux du plateau Mackenzie, nous avons fait une campagne à bord d'un navire en aûst-septembre 1990 (CCGS *Henry Larsen*, Institute des Sciences de la Mer, I.D. #9070), mené une campagne d'échantillonnage sur la glace au printemps 1991 (23 avril - 11 mai, Institute des Sciences de la Mer I.D. #9109). Les mesures physiques et chimiques obtenues lors de chacune des ces campagnes ont été rapportées antérieurement [*Macdonald et al.*, 1991, 1992]. Dans le présent rapport, nous avons colligés les résultats de la composition isotopique de l'oxygène ($\delta^{18}O$) et de la salinité des échantillons d'eau prélevés en septembre 1990 (campagne #9070) et des carottes de glace et des échantillons d'eau prélevés en avril - mai 1991 (campagne #9109).

Mots-clés: Arctique, $\delta^{18}O$, glace, zone côtière.

Acknowledgements

This work was funded by **Indian and Northern Affairs, Canada**, as part of the *Northern Oil and Gas Action Program*. We thank Frozen Sea group for the loan of equipment enabling us to work from the ice. Jimmy and Jackie Jacobson of Tuktoyaktuk helped us with the snowmobile sampling off Cape Bathurst. We are grateful for the support provided by Polar Continental Shelf Project, and in particular Barry Hough and Claude Brunet at Tuktoyaktuk. We thank the pilots, Ron Sprang and Pierrette Paroz, for getting us to and from the ice safely. Sample handling and preparation during the spring work was greatly facilitated by having the use of the Winnipeg DFO laboratory at Tuktoyaktuk; we thank G. Lacho for making this space available to us. We are very much indebted to the officers and men of the Canadian Coast Guard Ship *Henry Larsen*. We appreciate the special efforts of Captain S. Gomes, Captain D. Johns and Ivan Côté to provide ship time for the late summer programs. S. Thomson assisted with advice on style, and with final text-editing of this report.

1 INTRODUCTION

The Northern Oil and Gas Action Program has as one of its sub- projects a major inter-disciplinary study of the oceanography of the Canadian Beaufort Sea (NOGAP B.6). In particular, the objectives of NOGAP B.6 are to determine the transport and fate of materials (especially hydrocarbons) over the Beaufort Shelf, and the primary productivity of these coastal waters. Field work started in 1986 and the program was finally completed in March, 1994. Data reports in the NOGAP B.6 series are listed on the inside of the back cover of this report. Here we provide a description of the sampling and analytical methods for oxygen isotope ($\delta^{18}O$) measurements made on water and ice cores collected during two of the NOGAP B.6 missions. For other supporting data the reader is referred to *Macdonald et al.*, [1991, 1992]. Included in these earlier reports are CTD data and bottle data for measurements of salinity, temperature, nutrients (silicate, phosphate and nitrate), dissolved oxygen and chlorophyll a determinations. An interpretation of the September, 1990 and April-May 1991 oxygen isotope data is available in *Macdonald et al.*, [1995].

1.1 August-September, 1990, Institute of Ocean Sciences Mission #9070

This work was carried out from the CCG Icebreaker *Henry Larsen*. Objectives of the program and oceanographic data are described in detail elsewhere [*Macdonald et al.*, 1991]. Briefly, the objectives with respect to the $\delta^{18}O$ data reported here were:

- Collect chemical and CTD data at a deep station in the Canada Basin.
- Collect chemical and CTD data along a time-series transect across the Mackenzie Shelf.
- Collect comparative chemical and CTD data from the shelf west of Banks Island.
- Intensively survey the waters of the Mackenzie Canyon.

Figures 1 and 2 show the station locations.

1.2 April-May, 1991, Institute of Ocean Sciences mission # 9109

During this mission, our field work focused on the nearshore zone and the disposition of fresh water from the Mackenzie River under landfast ice in late winter. A preliminary study ([*Macdonald and Carmack*; 1991] led us to conclude that this nearshore region and this time of year are critical both to the biology and to physical processes that transport properties including contaminants during winter. Accordingly, we collected CTD data and water samples for chemistry on transects outward from the coast. Additionally, we collected ice cores at many of the stations. These cores were sectioned and analyzed for salinity, $\delta^{18}O$, and nutrients.

We operated the spring program out of Tuktoyaktuk, Polar Continental Shelf Project (PCSP), and used fixed-wing and rotary- wing support vehicles to carry out the sampling. During the spring of 1991, a particularly large polynya opened up off the Tuktoyaktuk Peninsula; to take a station in the open water we used a Zodiac workboat. Figure 3 shows the station locations for this mission.

Objectives of mission #9109 and the oceanographic data are described in [*Macdonald et al.* 1992]. The logistic goals accomplished with respect to oxygen isotope sampling were as follows (data reported here are in **bold** font:

- Collect water column samples in late winter for temperature, nutrients, dissolved oxygen, **salinity and $\delta^{18}O$** determination on transects extending outward from shore, covering especially region invaded by the Mackenzie River plume under the ice.

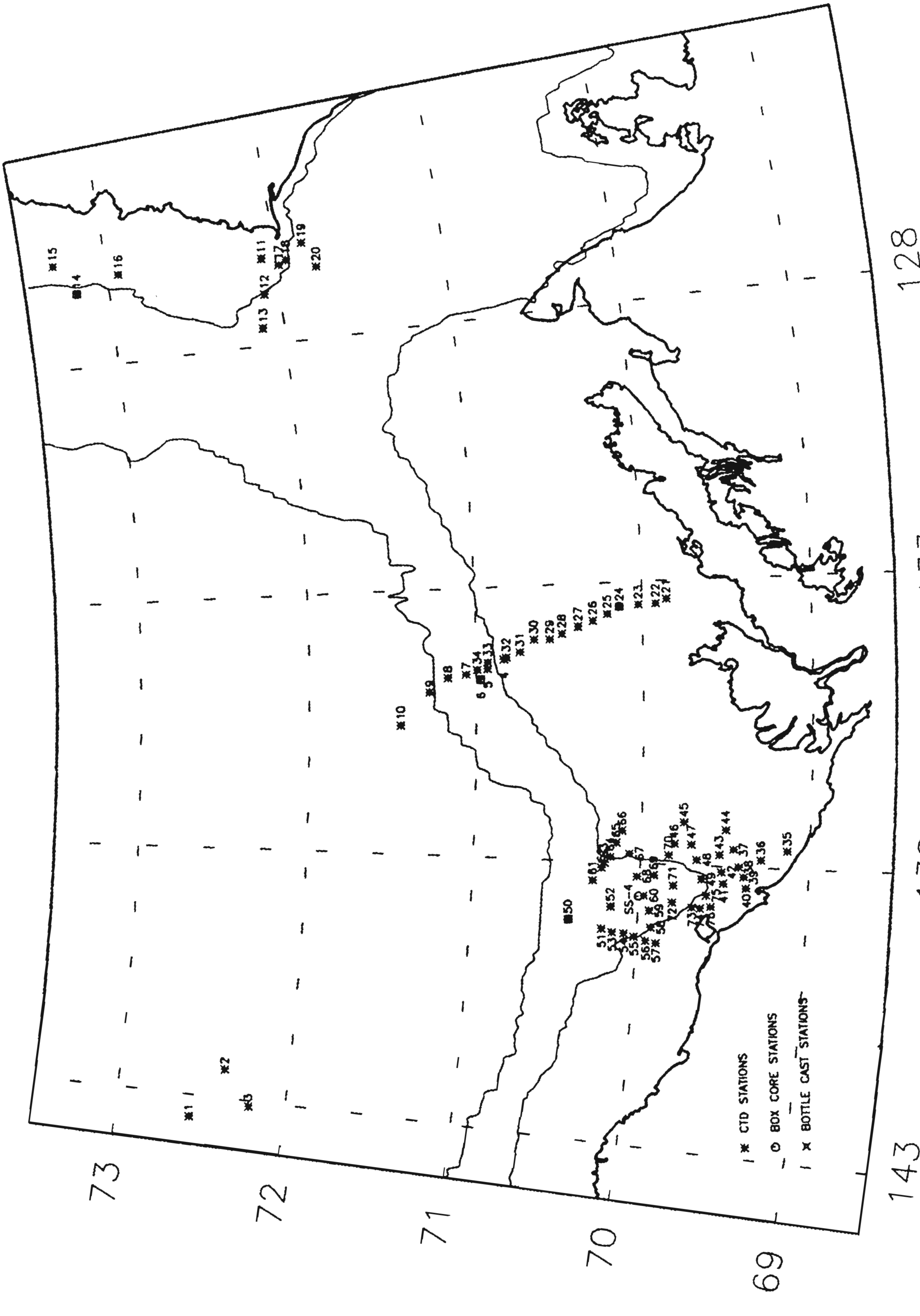

Figure 1: Station locations for August-September, 1990 (# 9070)

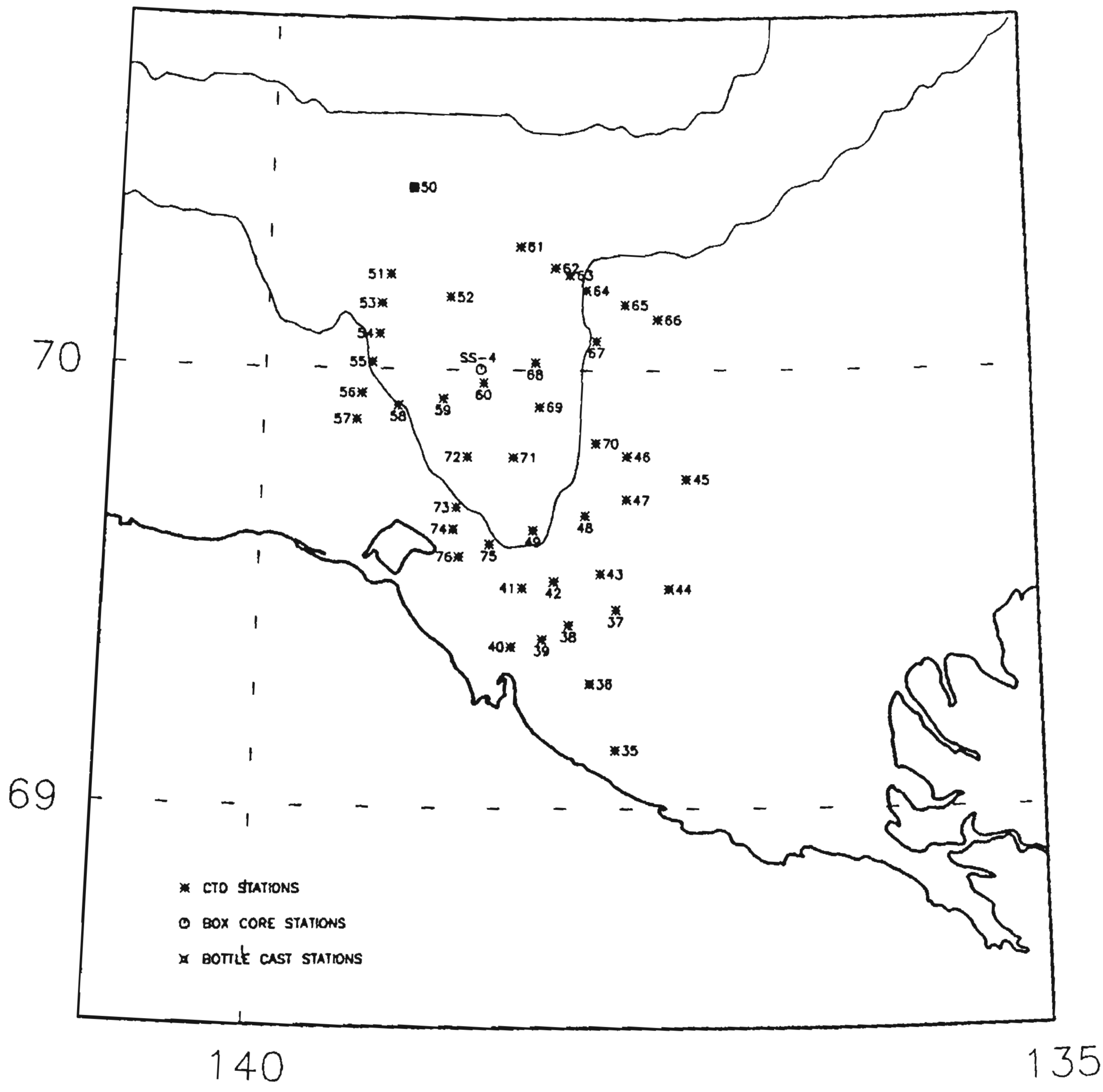

Figure 2: Station locations for Mackenzie Canyon survey, 1990 (# 9070)

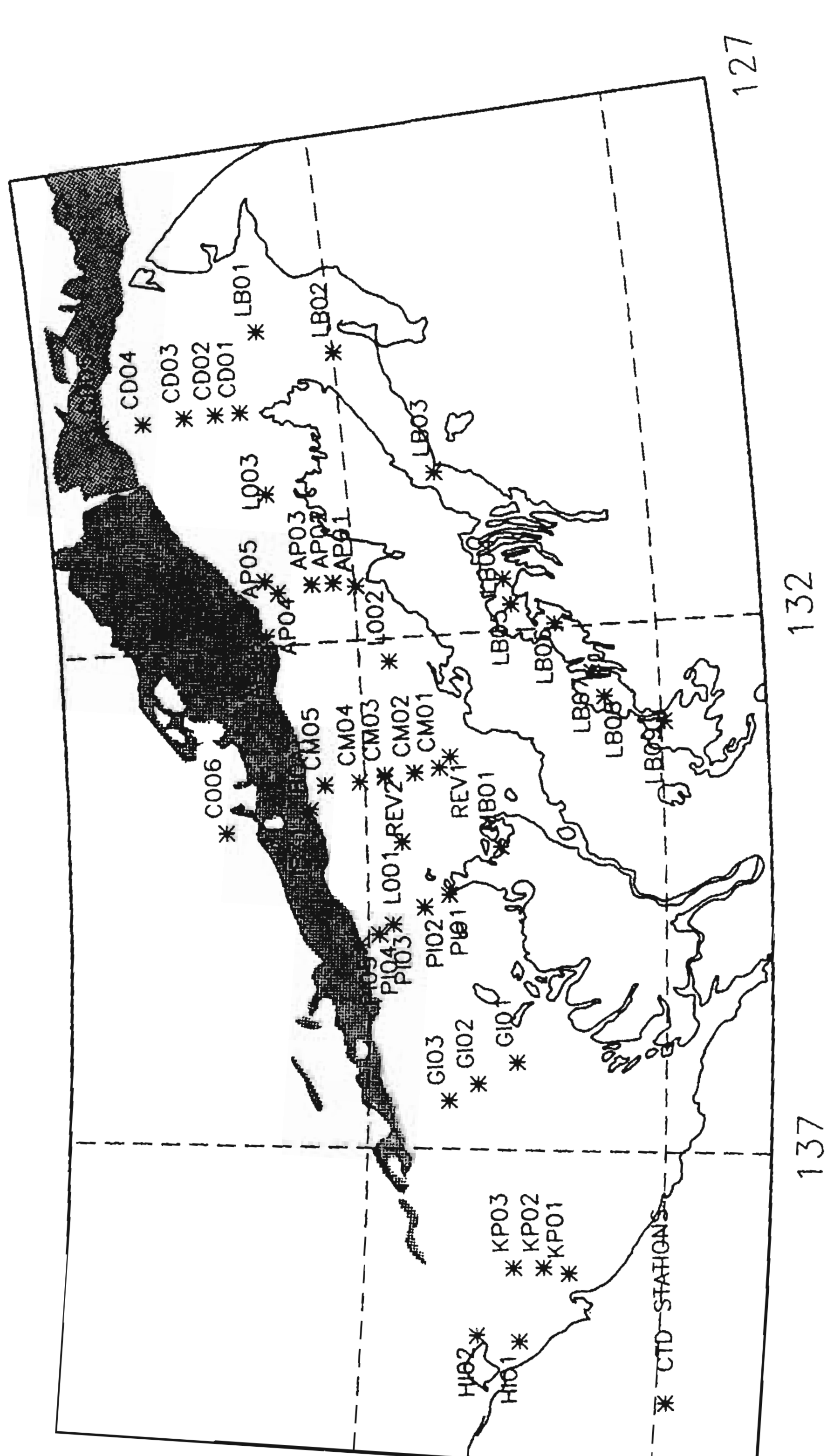

Figure 3: Station locations for April-May, 1991 (# 9109) – approximate position of open water is shown.

- Collect ice cores at most of the water sampling stations and at two Mackenzie River stations for which ice-thickness records were being collected and measure these for δ^{18}**O, salinity** and nutrients.

1.2.1 Station Nomenclature for Mission #9109

The stations have been given a two-part designation. The alphabetic before the hyphen refers generally to a coastal feature relevant to the location of the transect (described below) and, after the hyphen, stations are generally numbered sequentially outward from the coast.

HI Herschel Island

KP Kay Point

GI Garry Island

PI Pullen Island

CM C section off Tuktoyaktuk, Middle part

CI C section off Tuktoyaktuk, Inner part

C C section off Tuktoyaktuk, Outer part

AP Atkinson Point

CD Cape Dalhousie

L, SL Along-coast, stations on the 10 m isobath

CB Cape Bathurst

LB Liverpool Bay

REV Recording Expendable Velocimeter Station, located on the C transect.

MB Mason Bay

ISZ, TOW, BOAT Stations in the open lead

CAN, KIT, WSTC, West C, flgp, tkak Mackenzie River, Kugmallit Bay

CASI Calibration stations in the open lead

2 METHODS

2.1 Station Locations

Stations taken from the CCGS *Henry Larsen* were navigated with a Trimble "Trimpack" GPS (Global Positioning System) using C/A code and verified occasionally with the ship's transit Satellite Navigator. The status of Selective Availability is not known for this period. For GPS, the positions are expected to be accurate within about 50 m; for the transit Satellite Navigator, the majority of positions are expected to be within 930 m of the true position, with a mean displacement of about 476 m [*Huggett and Mortimer*, 1971].

For the ice-based work, all stations were navigated using the aircraft GPS, or for stations taken by snowmobile a Magellan hand-held GPS was used, again with C/A codes for positioning. During this field trip, the GPS "selective availability" was turned off and therefore positions are expected to have a reliability of about ±50m. In the case of this field work, precision of the GPS was repeatedly demonstrated by our ability to navigate back to single, unmarked auger holes in the landfast ice zone.

2.2 Field Sampling

2.2.1 Shipboard operations from the CCGS *Henry Larsen*

Sampling was carried out from the forward port side of the ship. Two winch pads and A-frame pads were used to mount Institute of Ocean Sciences (IOS) winches and A-frame transferred from the CCGS *George Pearkes*. The A-frame was used to suspend the block for over-the-side operations. One half of a heated container was used as the wet-lab where Niskin bottles were subsampled and thermometers were read; the other half, which was partitioned by a wall, was dedicated to the freon (chlorofluorocarbon) analyses. This container was mounted directly on the starboard side of the forward deck.

Salinity samples were drawn into 200 mL salinity bottles after 3 rinses from Niskin bottles or the pumping system. The samples were then capped tightly and care was taken to avoid freezing during sampling or transport. Samples for oxygen isotope determination were drawn into 30 mL CPE bottles after 3 rinses with sample water.

2.2.2 Ice-based operations

The general plan was to sample along pre-chosen transects distributed across the landfast ice zone, but stations were also occupied beyond this region. Sampling equipment and personnel were flown from PCSP in Tuktoyaktuk to the selected site by fixed wing aircraft (Twin Otter) or helicopter (Bell 206L Long Ranger). The actual site was for collecting water or ice cores was selected carefully from the air. We chose broad, flat regions of first-year ice that were well away from ridges or complex ice topography. Sampling equipment included a hand winch, 1.7 L Niskin sampling bottles as well as submersible pump systems for water and a Sipre ice corer for collecting ice.

Water sampling The pumping system consisted of a 316 stainless steel magnetically coupled pump with Ryton gears coupled to a submersible well pump motor (Franklin Electric Co., 3450 RPM). Water was pumped through a 1.4 cm o.d. hose (Aeroquip 2807-8) constructed of smooth bore extruded Teflon TFE (1.0 cm i.d.) with a reinforcement and cover of one-braid, high tensile stainless steel wire. The pump delivered approximately 6 L/min, and the hose length (max depth) was 50 m.

On the ice, a 25 cm hole was first augered and, when required by weather, a tent was placed over the hole. Bottle sampling followed standard oceanographic procedures. One depth was sampled at a time and subsampling from the bottles followed the order dissolved oxygen, salinity, nutrients, oxygen isotopes and Chl *a*. Water samples for oxygen isotope determination were stored in 30 mL CPE bottles after 3 rinses from the sample water. Salinity samples were drawn into 200 mL salinity bottles after 3 rinses from Niskin bottles or the pumping system. The samples were then capped tightly and care was taken to avoid freezing during sampling or transport. When the pump was used, a dilute ethanol-water mixture was kept in the hose to prevent freezing during transport and storage. Therefore, the hose was flushed at depth for 6 minutes before collecting the first sample at a station, and for 3 minutes at each specific depth after that. Samples were stored in an insulated box (with optional heating) to protect them from freezing and light; these were then shipped back to the Laboratory at Tuktoyaktuk on the same day (1-6 hours) when aircraft were used, and the next day when the snowmobile was used (Cape Bathurst). Chl *a* samples were collected into 1 L or 500 mL polyethylene bottles. Oceanographic thermometers were not used due to the difficulty of manipulating them through an ice hole, and the rough treatment they would receive in the field. Temperatures are generally available from CTD casts carried out at the same location.

Ice sampling Ice cores were collected using a hand-operated Sipre corer. The depth of snow was measured at several places in the vicinity of the site, snow was cleared from the site, and the corer was used to collect ice segments in lengths of about 70 cm. Discontinuities in the ice were noted during collection. The total length of core collected was carefully reconciled with the depth of the hole after each segment. This procedure was found to be particulary important for river ice which was brittle and tended to fracture easily during the coring process. The ice cores were immediately cut with a saw into 10 cm sections. For the cutting, a half tube mounted in a frame was used to avoid contact of the core with the ice surface to avoid contamination of the sections with snow. Each cut section was placed immediately into a labelled plastic container which was sealed with a screw cap. Coring was continued until water was reached and at the end of this process, a sample of water from just beneath the ice was collected using a tube inserted through the cored hole. The containers were returned to the laboratory in Tuktoyaktuk where the contents were allowed to thaw at room temperature. The liquid was then homogenized and subsampled for salinity, $\delta^{18}O$, and nutrient determinations.

2.3 Laboratory Methods

2.3.1 $\delta^{18}O$ determination

5.0 mL of sample water (seawater or melted ice) were pipetted into a 25 $\times$ 150 mm culture tube. A micro stir bar was added to each sample tube and the tubes were secured by Vac-Torr fittings into a radially configured equilibration chamber with 16 sample positions [*Whaite*, 1982]. Each sample tube was stirred by four small electromagnets surrounding each test tube providing a switching magnetic field which powers the stirring process. A VWR Scientific refrigerated recirculator set at 16 °C, provided a constant temperature of 20 $\pm$ 0.5 °C in the equilibration chamber. Software written in- house controlled the electronically activated solenoids which opened and closed valves as programmed for the equilibration process. A multi-tasking, MS-DOS computer controlled the sample equilibration, mass spectrometer inlet system valve control and data acquisition.

The mass spectrometer, built by Nuclide, has a 5 inch radius of curvature with a 45 ^{o} deflection. The normal operating conditions were: 4.5 A filament current, 0.2 mA trap current, 0.8 MA shield, 0 V repellor, 65 V electrometer accelerating potential and 4.1 kV ion accelerating potential. The instrument is a triple collector system equipped with evacuated electrometer heads (Nuclide EAH-500 Faraday cup detectors). Under normal operating conditions there was about 4 cm of CO_2 pressure on the high pressure side of the viscous leaks which resulted in a mass 44 ion beam intensity of 2.0 $\times$ 10-8 A. The introduction of gas samples into the mass spectrometer was accomplished with a dual gas inlet system with the ion beam intensities being balanced by manually adjusted bellows. We analyzed two of the standard water samples listed in the Table, one empty tube and one duplicate sample with each batch of 16 samples. Software programs to control the equilibration process, inlet system valve control and integrating ratiometer data acquisition on the mass spectrometer for the analysis were written in Quick Basic and are included here in the appendix.

The following steps were involved in this computer-controlled equilibration process.

1. All reservoir lines were opened, evacuated and tested for leaks to the atmosphere.
2. Transfer lines, reservoirs and sample tubes were evacuated and approximately 2.0 psi of CO_2 was introduced into the sample test tubes.
3. Fifteen hours was allowed for equilibration with CO_2, during which oxygen in the CO_2 standard gas (Matheson Research grade) exchanges with the oxygen in water through the following equilibrium:

$$CO_2 + H_2O \rightleftharpoons H_2CO_3$$

4. After equilibration, the sample reservoirs were evacuated and transfer lines closed automatically.
5. The equilibrated CO_2 head gas in each sample tube was allowed to expand into the reservoir and was then sealed and held awaiting the manual introduction of the gas sample to the mass spectrometer inlet system by the operator.
6. Each sample was passed through a Peltier cooler operating at about -25 °C during transfer to the inlet system. The Peltier cooler froze out and trapped any water vapour in the CO_2 gas as it passed through the transfer line to the sample side of the mass spectrometer inlet system.
7. At the mass spectrometer, gas from the same CO_2 tank used for the equilibration of the sample was used as the reference gas for the analysis. The pressure of the two gases was balanced using bellows in the sample inlet system.
8. The capillary valves were opened by computer control to allow alternating aliquots of sample and reference gas to enter the flight tube. Data acquisition parameters were user- selectable; here, we employed seven reference/sample gas cycles, each cycle comprising an average of seven ratio readings. The software allows the operator to modify the screening criterion so that statistically rejected (Chauvenet's criterion) outlying data points can be dropped from the data.

Results are reported relative to Vienna Standard Mean Ocean Water (V-SMOW) as $\delta^{18}O$ where:

$$\delta^{18}O = \left[\frac{(^{18}O/^{16}|rmO)_{sample}}{(^{18}O/^{16}O)_{V-SMOW}} - 1\right] \times 1000$$

We used V-SMOW standard plus in-house standard water samples intercalibrated through the University of Washington (Table 1)

Initially, we allowed the equilibrator temperature to float with the room temperature but during particularly warm days there was a sufficient fluctuation in the room temperature to produce an offset in the results. We therefore installed an air conditioner to control equilibrator temperature. This improved the stability but unacceptable variation was still observed as the inlet water and room temperature fluctuated. We therefore modified the equilibration chamber to provide evenly-distributed air circulation ports and installed a closed system with a forced air heat exchanger controlled through a constant-temperature recirculating bath. This system maintains the equilibrator at a constant temperature of $\pm 0.5C^{\circ}$.

We have processed the data and error statements for each batch of analyses separately so there are three separate error statements presented here.

Table 1: Composition of Reference and Control Water Samples

Water	$\delta^{18}O$	Precision
V-SMOW	0.00	
Institute of Ocean Sciences	-9.74	0.06
Tuktoyaktuk Snow	-27.17	0.07
University of British Columbia-LTW	-16.62	0.03
University of Washington (Antarctic)	-33.46	

Table 2: Summary of calibration data for the period December 1991 to July 1992

X_{calc}	Y value	95% C.I.	Replicates
	(V-SMOW)	(k)	(n)
26.0	+2	0.376	1
		0.271	2
		0.225	3
14.0	+14	0.372	1
		0.265	2
		0.218	3
1.0	+27	0.395	1
		0.297	2
		0.255	3

Treatment of Error, Control Charts The V-SMOW $\delta^{18}O$ values were determined through the use of a linear regression model for mixtures of in-house standards [*Macdonald and O'Brien*, 1985] which have been intercalibrated against V-SMOW by the University of Washington - Quaternary Research Centre. This error model provides more security in that it allows one to trace a measurement to two or more references simultaneously.

The 9109 ice core dataset was run from December 1991 to July 1992. The accuracy of the calibration curve over the calibration range is summarized in Table 2 and the conrol chart for this period is given in Figure 4.

During this period the pooled standard deviation for same day duplicate analysis of samples was 0.18 ($n = 46$). The laboratory temperature remained fairly uniform over the period of analysis and the 95% C.I. was slightly wider than we were able to obtain later with the constant-temperature bath.

The 9109 water column dataset was run during the period of July 1992 to August 1992. The accuracy of the calibration curve over the calibration range is summarized in Table 3 and the control chart is given in Figure 5. During this period, the pooled standard deviation for same day duplicate analysis of samples was 0.08 ($n = 21$). Control of the laboratory temperature was poor due to summer heating and subsequent over-cooling from the air conditioner which was installed. As noted above, a temperature controller was installed to address this problem.

The 9070 water column dataset was run from May 1993 to July 1993. The accuracy of the calibration curve over the calibration range is given in Table 4 and the control chart in Figure 6. The pooled standard deviation for same day duplicate analysis of samples was 0.02 (n=44). The

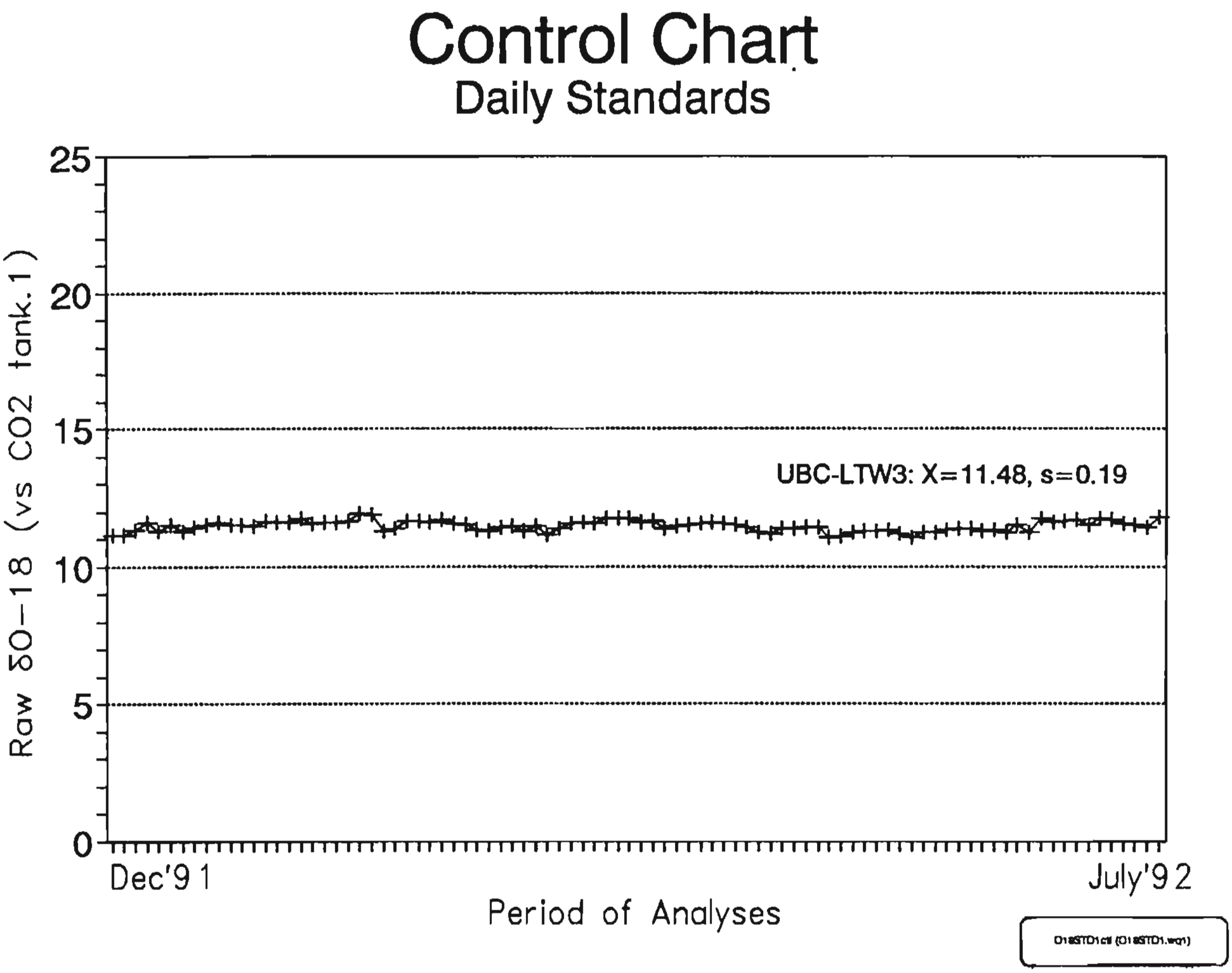

Figure 4: Control chart for the period December, 1991 to July, 1992

Table 3: Summary of calibration data for the period July 1992 to August 1992

X_{calc}	Y value	95% C.I.	Replicates
	(V-SMOW)	(k)	(n)
26.0	-13	0.76	1
		0.55	2
		0.46	3
13.7	-7	0.75	1
		0.54	2
		0.45	3
0.5	+6	0.79	1
		0.59	2
		0.51	3

Table 4: Summary of calibration data for the period May 1993 to July 1993

X_{calc}	Y value	95% C.I.	Replicates
	(V-SMOW)	(k)	(n)
26.0	-20	0.223	1
		0.159	2
		0.131	3
13.7	-8	0.223	1
		0.158	2
		0.130	3
0.4	+5	0.226	1
		0.163	2
		0.136	3

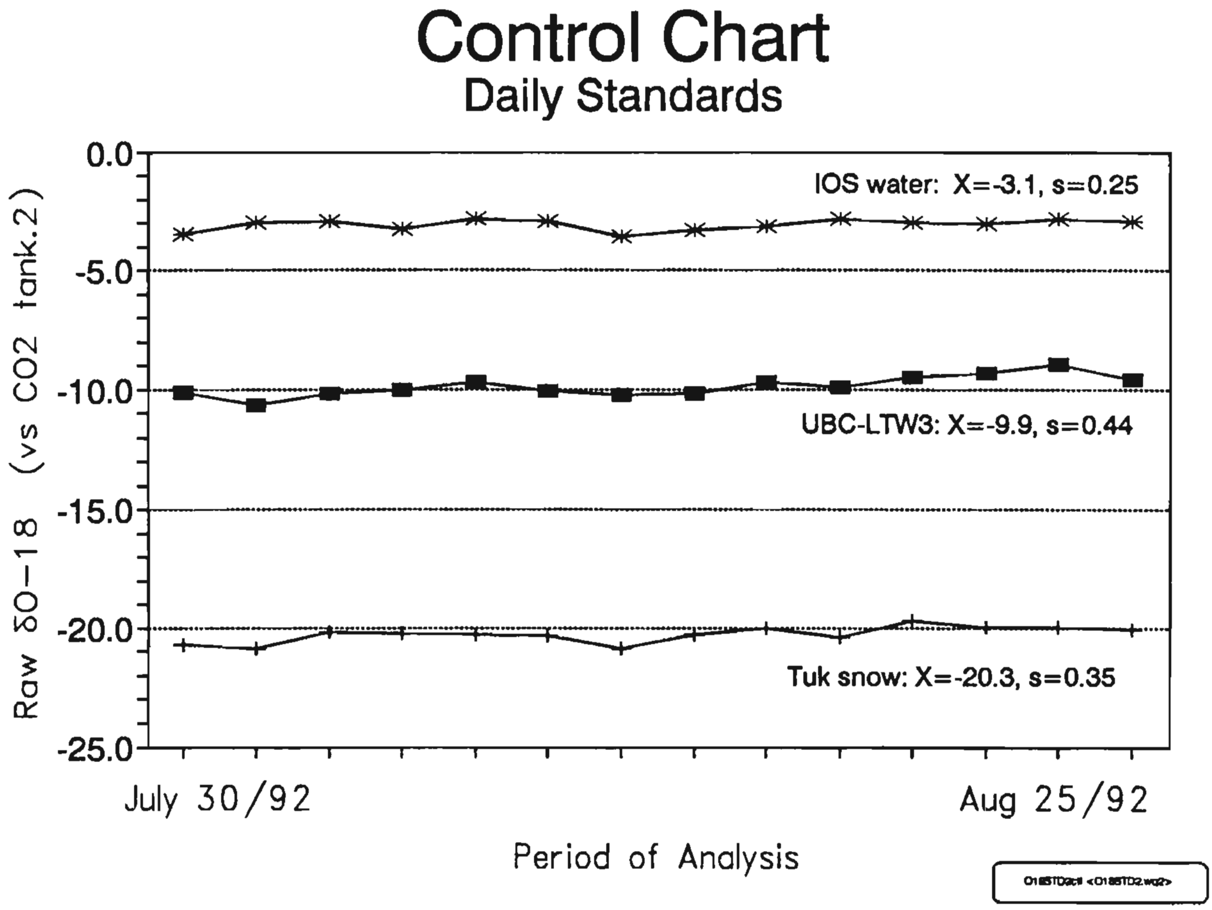

Figure 5: Control chart for July 30, 1992 to August 25, 1992

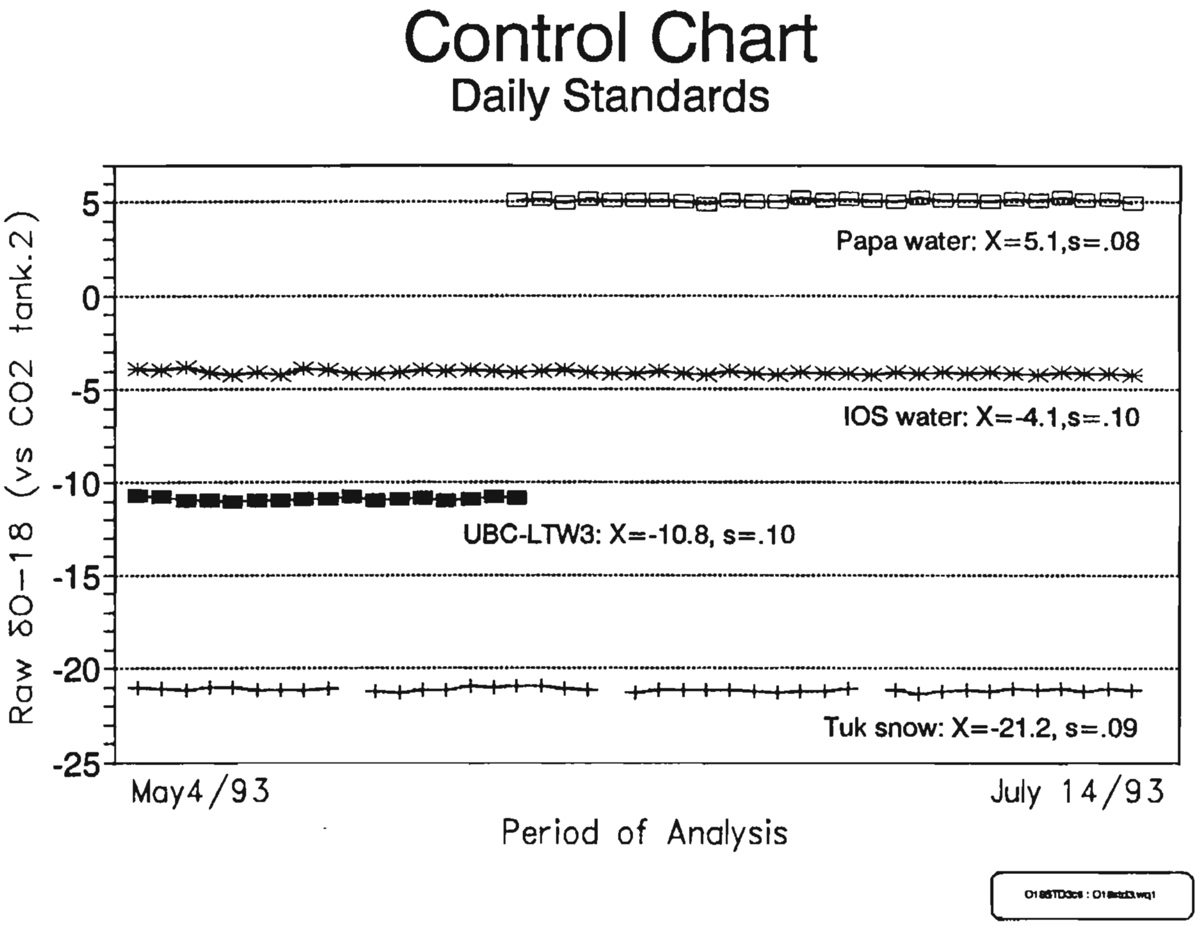

Figure 6: Control chart for May 4, 1993 to July 14, 1993

isothermal, forced-air modification installed on the equilibrator maintained temperatures to ±0.2° C over the period of analysis significantly improving the 95% C.I. over the previous two analytical periods.

2.3.2 Salinity determination

The salinities were analyzed at the DFO laboratory in Tuktoyaktuk or at IOS on a Guildline Autosal (Model 8400A) instrument as described previously [*Macdonald et al.*, 1991, 1992]; data are reported in practical salinity units (psu) [see *Lewis and Perkin*; 1978]. Instrumental precision determined from repeated analyses on the same sample was about ±0.003. During analyses the instrument was standardized against Standard Sea Water obtained from the Standard Seawater Service, Institute of Oceanography, Wormley, Godalming, Surrey, England. Pooled variance, s_p, is

Table 5: Error Summary for Salinity Determinations

Mission #	Precision	ν	Reference Water
9070	0.0066	32	IAPSO (K_{15} = 0.99984)
9109	0.016	7	IAPSO K_{15} = 0.99986

calculated as:

$$s_p = \sqrt{\frac{\nu_1 s_1^2 + \cdots + \nu_i s_i^2}{\nu_1 + \cdots + \nu_i}}$$

where $\nu_i = n_i - 1$ degrees of freedom, and the n_i and s_i refer to the number of replicates and their standard deviation for the individual components used in the pooled standard deviation calculation. Table 5 shows the estimated error (from *Macdonald et al.* [1991, 1992]) for the salinity determinations

3 References

Huggett, W.S. and A. Mortimer, 1971. Observations obtained on a Magnavox Satellite Navigation Receiver in high latitudes. *Pacific Marine Science Report*, **71-4**, 13pp.

Lewis, E.L. and R.G. Perkin, 1978. The practical salinity scale 1978: conversion of existing data. *Deep Sea Res.*, **28A**, 307-328.

Macdonald, R.W., and E.C. Carmack, 1991. The role of large-scale under-ice topography in separating estuary and ocean on an Arctic shelf, *Atmosphere-Ocean*, **29**, 37-53.

Macdonald, R.W., E.C. Carmack, F.A. McLaughlin, D. Sieberg, M. C. O'Brien, D. Paton, R. Pearson, Yang Liangfeng, C. Gobeil, 1991, Oceanographic Data Collected from the *Henry Larsen* in the Beaufort Sea, August-September 1990. *Can. Data Rep. Hydrogr. Ocean Sci.*: **97**, 142 pp.

Macdonald, R.W., and M.C. O'Brien, 1985. Extending the use of certified reference sediments for assessment of accuracy in determinations of trace metals, *Anal. Chim. Acta*, **177**, 81-91.

Macdonald, R.W., D.W. Paton, E.C. Carmack, A. Omstedt, 1995. The freshwater budget and under-ice spreading of Mackenzie River water in the Canadian Beaufort Sea based on salinity and $^{1}8O/^{1}6O$ measurements in water and ice, *J. Geophys. Res.*, 100 (C1), 895-919.

Macdonald, R.W., R. Pearson, D. Sieberg, F.A. McLaughlin, M.C. O'Brien, D. Paton, E.C. Carmack, J.R. Forbes, J. Barwell- Clarke, 1992. NOGAP B.6, Physical and Chemical Data Collected in the Beaufort Sea and Mackenzie River delta, April-May 1991 *Can. Data Rep. Hydrogr. Ocean Sci.*: **104**, 155 pp.

White, P., An automated sample line for the preparation fo $^{18}O/^{16}O$ isotope analyses from water samples, M.Sc. Thesis, University of British Columbia, Vancouver, B.C., 165pp.

4 APPENDIX A, DATA TABLES

4.1 Mission #9070

STATION A-01
CRUISE 9070

DATE 31/8-04/09/90
TIME (Z-6)

LATITUDE (DEG MIN) 72 35 N
LONGITUDE (DEG MIN) 143 28 W

WATER DEPTH (m) 3325

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L1	0	21.525	-4.73
L2	10	26.437	-3.94
L3	20	30.066	-4.09
L4	30	30.378	-3.98
L5	40	30.650	-3.87
L6	50	31.367	-3.51
L7	60	31.669	-3.14
L8	70	31.963	-2.67
L9	80	32.256	-2.36
L10	100	32.489	-1.98
L11	117	32.605	-1.94
L12	136	32.699	-1.78
L13	146	32.750	-1.66
L14	156	32.816	-1.73
L15	166	32.855	-1.70
L16	175	32.897	-1.62
L17	185	32.947	-1.65
L18	195	32.983	-1.67
L19	219	33.266	-1.73
L20	244	33.739	-1.33
L21	257	34.042	-1.06
L22	280	34.406	-0.60
L23	303	34.587	-0.28
L24	327	34.664	-0.17
L25	350	34.673	-0.01
L26	373	34.757	-0.00
L27	420	34.808	0.03
L28	467	34.827	-0.03
L29	513	34.850	0.03
L30	560	34.861	0.13
L31	700	34.870	-0.02
L32	800	34.885	0.07
L33	900	34.893	0.02
L34	1000	n.g.	0.04
L35	1250	34.915	0.10
L36	1500	34.917	0.05
L37	1600	34.927	0.09
L38	1700	34.938	0.10
L39	1800	34.941	0.13
L40	1900	34.940	0.09
L41	2000	34.954	0.06
L42	2250	34.958	0.06
L43	2500	34.947	0.17
L44	3000	34.961	0.06
L46	3200	34.957	0.03
L47	3300	34.961	0.10

STATION			L-11		
CRUISE			9070		
DATE			11/9/90		
TIME (Z-6)			1530		
LATITUDE	(DEG	MIN)	72	5.2	N
LONGITUDE	(DEG	MIN)	126	20.2	W
WATER DEPTH (m)			23		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L48	0	28.798	-3.73
L49	5	28.851	-3.62
L50	10	29.138	-3.51
L51	20	30.989	-3.34

STATION			L-12		
CRUISE			9070		
DATE			11/9/90		
TIME (Z-6)			1354		
LATITUDE	(DE	MIN)	72	5.6	N
LONGITUDE	(DE	MIN)	127	1.99	W
WATER DEPTH (m)			110		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L52	0	28.243	-3.58
L53	5	28.251	-3.61
L54	10	29.359	-3.54
L55	20	30.957	-3.30
L56	30	31.542	-3.21
L57	50	32.464	-2.33
L58	75	32.810	-1.99
L59	100	33.013	-1.84

STATION	L-13
CRUISE	9070
DATE	11/9/90
TIME (Z-6)	1827
LATITUDE (DEG MIN)	72 7.87 N
LONGITUDE (DEG MIN)	127 38.02 W
WATER DEPTH (m)	370

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L60	0	25.098	-3.99
L61	5	27.941	-3.75
L62	10	28.023	-3.86
L63	20	30.278	-3.45
L64	30	31.306	-3.25
L65	50	31.826	-3.11
L66	75	32.264	-2.30
L67	100	32.720	-1.86
L68	125	33.021	-1.48
L69	150	33.333	-1.58
L70	200	34.105	-0.90
L71	250	34.576	-0.18
L72	300	34.739	-0.15
L73	350	34.795	-0.03

STATION	L-14
CRUISE	9070
DATE	12/9/90
TIME (Z-6)	838
LATITUDE (DE MIN)	73 11.55 N
LONGITUDE (DE MIN)	126 29.28 W
WATER DEPTH (m)	112

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L74	0	28.773	-3.55
L75	5	28.749	-3.67
L76	10	28.750	-3.59
L77	20	30.118	-3.45
L78	30	31.650	-3.10
L79	50	32.184	-2.45
L80	75	32.632	-1.78
L81	100	32.945	-1.76

STATION			L-15		
CRUISE			9070		
DATE			12/9/90		
TIME (Z-6)			1223		
LATITUDE	(DEG	MIN)	73	19.25	N
LONGITUDE	(DEG	MIN)	125	48.6	W
WATER DEPTH (m)			81		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L82	0	29.777	-3.51
L83	5	29.777	-3.60
L84	10	29.825	-3.59
L85	20	broken	-3.56
L86	30	31.161	-3.22
L87	50	31.991	-2.88
L88	75	32.502	-2.14

STATION			L-16		
CRUISE			9070		
DATE			12/9/90		
TIME (Z-6)			1506		
LATITUDE	(DE	MIN)	72	55.96	N
LONGITUDE	(DE	MIN)	126	12.85	W
WATER DEPTH (m)			43		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L89	0	29.645	-3.60
L90	5	29.662	-3.72
L91	10	29.758	-3.57
L92	20	30.623	-3.56
L93	30	31.770	-3.06

STATION			L-21		
CRUISE			9070		
DATE			14/9/90		
TIME (Z-6)			920		
LATITUDE	(DEG	MIN)	69	51.11	N
LONGITUDE	(DEG	MIN)	133	19.17	W
WATER DEPTH (m)			15		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L94	0	12.938	-12.06
L95	5	28.100	-4.70
L96	10	29.261	-3.95

STATION			L-22		
CRUISE			9070		
DATE			14/9/90		
TIME (Z-6)			1030		
LATITUDE	(DE	MIN)	69	55.28	N
LONGITUDE	(DE	MIN)	133	23.39	W
WATER DEPTH (m)			21		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L97	0	20.679	-8.34
L98	5	28.727	-4.30
L99	10	29.224	-4.20
L100	15	29.410	-4.03

STATION			L-23		
CRUISE			9070		
DATE			14/9/90		
TIME (Z-6)			1138		
LATITUDE	(DEG	MIN)	70	1.63	N
LONGITUDE	(DEG	MIN)	133	24.43	W
WATER DEPTH (m)			31		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L101	0	28.760	-4.23
L102	5	28.999	-4.05
L103	10	29.278	-3.93
L104	20	30.861	-3.32
L105	25	30.999	-3.18

STATION			L-24		
CRUISE			9070		
DATE			14/9/90		
TIME (Z-6)			1252		
LATITUDE	(DE	MIN)	70	8.3	N
LONGITUDE	(DE	MIN)	133	26.03	W
WATER DEPTH (m)			41		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L106	0	28.115	-3.66
L107	5	28.110	-3.79
L108	10	28.421	-3.70
L109	20	31.158	-3.02
L110	30	31.574	-2.84

STATION L-25
CRUISE 9070

DATE 14/9/90
TIME (Z-6) 1520

LATITUDE (DEG MIN) 70 13.12 N
LONGITUDE (DEG MIN) 133 33.73 W

WATER DEPTH (m) 50

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L111	0	28.039	-3.83
L112	5	28.043	-3.71
L113	10	28.667	-3.91
L114	20	31.298	-3.42
L115	30	31.653	-3.18
L116	45	32.011	-2.71

STATION L-26
CRUISE 9070

DATE 14/9/90
TIME (Z-6) 1647

LATITUDE (DE MIN) 70 18.2 N
LONGITUDE (DE MIN) 133 40.2 W

WATER DEPTH (m) 53

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L117	0	27.614	-3.80
L118	5	27.688	-3.73
L119	10	27.862	-3.70
L120	20	31.106	-3.38
L121	30	31.538	-3.12
L122	50	32.146	-2.55

STATION			L-27		
CRUISE			9070		
DATE			14/9/90		
TIME (Z-6)			1807		
LATITUDE	(DEG	MIN)	70	23.37	N
LONGITUDE	(DEG	MIN)	133	46.14	W
WATER DEPTH (m)			58		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L123	0	27.511	-3.91
L124	5	27.705	-3.77
L125	10	28.424	-3.86
L126	20	31.175	-3.39
L127	30	31.472	-3.38
L128	50	32.281	-2.37

STATION			L-28		
CRUISE			9070		
DATE			14/9/90		
TIME (Z-6)			1908		
LATITUDE	(DE	MIN)	70	29.1	N
LONGITUDE	(DE	MIN)	133	53.14	W
WATER DEPTH (m)			61		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L129	0	27.116	-3.72
L130	5	27.264	-3.92
L131	10	27.785	-4.00
L132	20	30.580	-3.67
L133	30	31.168	-3.66
L134	50	32.193	-2.62

STATION L-29
CRUISE 9070

DATE 15/9/90
TIME (Z-6) 840

LATITUDE (DEG MIN) 70 33.81 N
LONGITUDE (DEG MIN) 133 57.78 W

WATER DEPTH (m) 62

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L135	0	26.884	-3.79
L136	5	26.902	-3.70
L137	10	27.150	-3.70
L138	20	30.848	-3.68
L139	30	31.396	-3.44
L140	50	32.156	-2.58

STATION L-30
CRUISE 9070

DATE 15/9/90
TIME (Z-6) 1015

LATITUDE (DE MIN) 70 39.21 N
LONGITUDE (DE MIN) 133 59.47 W

WATER DEPTH (m) 65

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L141	0	26.926	-3.72
L142	5	26.934	-3.71
L143	10	27.001	-3.74
L144	20	30.784	-3.80
L145	30	31.289	-3.65
L146	50	32.144	-2.62

STATION L-32
CRUISE 9070

DATE 15/9/90
TIME (Z-6) 1240

LATITUDE (DEG MIN) 70 49.29 N
LONGITUDE (DEG MIN) 134 17.15 W

WATER DEPTH (m) 71

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L147	0	26.668	-3.81
L148	5	26.731	-3.66
L149	10	29.468	-3.82
L150	20	30.601	-3.66
L151	30	31.117	-3.70
L152	50	31.836	-3.07
L153	60	32.155	-2.75

STATION L-33
CRUISE 9070

DATE 15/9/90
TIME (Z-6) 1356

LATITUDE (DE MIN) 70 55.59 N
LONGITUDE (DE MIN) 134 23.68 W

WATER DEPTH (m) 107

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L154	0	26.436	-3.88
L155	5	26.440	-3.87
L156	10	26.502	-3.79
L157	20	30.041	-3.68
L158	30	30.777	-3.66
L159	50	31.636	-3.29
L160	75	32.432	-2.11
L161	100	32.665	-2.08

STATION	L-34
CRUISE	9070
DATE	15/9/90
TIME (Z-6)	1548
LATITUDE (DEG MIN)	70 59.09 N
LONGITUDE (DEG MIN)	134 29.78 W
WATER DEPTH (m)	265

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L162	0	26.276	-3.73
L163	5	26.292	-3.67
L164	10	26.446	-3.76
L165	20	30.439	-3.68
L166	30	30.814	-3.68
L167	50	31.655	-3.20
L168	75	32.287	-2.34
L169	100	32.596	-2.08
L170	125	32.776	-1.91
L171	150	32.948	-1.81
L172	184	33.267	-1.61
L173	234	34.186	-0.71

STATION	L-35
CRUISE	9070
DATE	16/9/90
TIME (Z-6)	837
LATITUDE (DE MIN)	69 8 N
LONGITUDE (DE MIN)	137 41.62 W
WATER DEPTH (m)	23

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L174	0	27.830	-4.02
L175	5	27.840	-4.09
L176	10	28.611	-3.79
L177	20	29.094	-3.93

STATION			L-36		
CRUISE			9070		
DATE			16/9/90		
TIME (Z-6)			957		
LATITUDE	(DEG	MIN)	69	16.9	N
LONGITUDE	(DEG	MIN)	137	51.02	W
WATER DEPTH (m)			37		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L178	0	26.595	-4.45
L179	5	26.521	-4.31
L180	10	26.538	-4.37
L181	20	28.794	-3.96
L182	30	30.473	-3.33

STATION			L-38		
CRUISE			9070		
DATE			15/9/90		
TIME (Z-6)			1204		
LATITUDE	(DE	MIN)	69	24.65	N
LONGITUDE	(DE	MIN)	137	58.6	W
WATER DEPTH (m)			52		

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L183	0	26.514	-4.33
L184	5	26.658	-4.35
L185	10	26.698	-4.14
L186	20	30.801	-3.35
L187	30	31.179	-3.14
L188	45	31.591	-2.77

STATION L-42
CRUISE 9070

DATE 16/9/90
TIME (Z-6) 1454

LATITUDE	(DEG	MIN)	69	30.85	N
LONGITUDE	(DEG	MIN)	138	4.47	W

WATER DEPTH (m) 92

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L189	0	24.933	-4.65
L190	5	25.235	-4.55
L191	10	25.406	-4.56
L192	20	31.170	-3.17
L193	30	31.666	-2.98
L194	50	31.762	-2.73
L195	75	31.903	-2.64
L196	85	32.119	-2.54

STATION L-49
CRUISE 9070

DATE 16/9/90
TIME (Z-6) 2134

LATITUDE	(DE	MIN)	69	37.95	N
LONGITUDE	(DE	MIN)	138	14.09	W

WATER DEPTH (m) 133

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L197	0	broken	-4.70
L198	5	24.815	-4.55
L199	10	27.228	-4.03
L200	20	30.536	-3.71
L201	30	31.105	-3.56
L202	50	32.063	-2.66
L203	75	32.355	-2.27
L204	100	32.466	-2.18
L205	125	32.719	-1.91

STATION	L-50
CRUISE	9070
DATE	17/9/90
TIME (Z-6)	827, 1055
LATITUDE (DEG MIN)	70 24.6 N
LONGITUDE (DEG MIN)	139 4.5 W
WATER DEPTH (m)	708

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L206	0	22.039	-4.88
L207	5	23.634	-4.42
L208	10	24.717	-4.26
L209	20	30.117	-3.82
L210	30	30.500	-3.82
L211	50	31.570	-3.23
L212	75	32.157	-2.55
L213	100	32.494	-2.20
L214	125	32.767	-1.95
L215	150	33.016	-1.83
L216	200	33.625	-1.52
L217	260	34.393	-0.52
L218	300	34.661	0.02
L219	350	34.750	0.23
L220	400	34.794	0.14
L221	450	34.829	0.21
L222	500	34.845	0.22
L223	550	34.857	0.25
L224	600	34.866	0.25
L225	650	34.871	0.17

STATION	L-52
CRUISE	9070
DATE	17/9/90
TIME (Z-6)	1435, 1510
LATITUDE (DE MIN)	70 9.65 N
LONGITUDE (DE MIN)	138 48.5 W
WATER DEPTH (m)	368

SEQ.NO	SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
L226	0	21.869	-5.44
L227	5	22.349	-5.63
L228	10	29.303	-3.94
L229	20	30.334	-3.81
L230	30	30.579	-3.69
L231	50	31.503	-3.30
L232	75	32.234	-2.46
L233	100	32.495	-2.22
L234	125	32.753	-1.94
L235	150	32.927	-1.90
L236	200	33.416	-1.68
L237	220	33.809	-1.27
L238	250	34.484	-0.39
L239	300	34.688	-0.12
L240	350	34.771	0.06

4.2 Mission #9109

STATION			HW-13		
CRUISE			9109		
DATE			25/04/91		
TIME ARRIVE (Z-6)			1500		
TIME DEPART (Z-6)			1630		
LATITUDE	(DEG	MIN)	68	7.67	N
LONGITUDE	(DEG	MIN)	134	27.92	W
ICE DEPTH (m)			0.71	**	
FREEBOARD (m)			-0.01		
SNOW DEPTH (cm)			29		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10		-16.94
10-20		-15.83
20-30		-15.49
30-40		-15.67
40-50		-15.80
50-60		-15.94
60-70		-15.82
70-80		-15.48
water		-18.22

** NB. The ice depth was only 0.71m but the core fractured and splintered so the core appeared to be 0.80 m.

STATION			HW-14		
CRUISE			9109		
DATE			25/04/91		
TIME ARRIVE (Z-6)			1100		
TIME DEPART (Z-6)			1230		
LATITUDE	(DEG	MIN)	68	26.25	N
LONGITUDE	(DEG	MIN)	133	48.58	W
ICE DEPTH (m)			1.5		
FREEBOARD (m)			0.06		
SNOW DEPTH (cm)			19		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10		-16.05
10-20		-15.69
20-30		-15.71
30-40		-15.62
40-50		-15.59
50-60		-15.79
60-70		-15.51
70-80		-15.59
80-90		-15.64
90-100		-15.50
100-110		-15.39
110-120		-15.27
120-130		-15.34
130-140		-15.69
140-150		-15.54
water		-18.14

STATION			PI-1		
CRUISE			9109		
DATE			26/04/91		
TIME ARRIVE (Z-6)			1712		
TIME DEPART (Z-6)			1828		
LATITUDE	(DEG	MIN)	69	43.84	N
LONGITUDE	(DEG	MIN)	134	31.25	W
ICE DEPTH (m)			1.95		
FREEBOARD (m)			0.13		
SNOW DEPTH (cm)			9		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	8.050	-4.79
10-20	5.579	-5.65
20-30	2.711	-9.62
30-40	1.830	-14.53
40-50	1.602	-14.75
50-60	1.822	-13.94
60-70	1.339	-15.49
70-80	1.249	-15.57
80-90	1.666	-14.77
90-100	1.012	-16.00
100-110	0.650	-16.32
110-120	0.311	-16.22
120-130	0.173	-16.12
130-146	0.074	-15.80
146-160	0.070	-15.97
160-170	0.012	-15.51
170-180	0.012	-15.74
180-190	0.032	-15.58
190-195	0.258	-15.79
water		-18.09

STATION			PI-2		
CRUISE			9109		
DATE			26/04/91		
TIME ARRIVE (Z-6)			1553		
TIME DEPART (Z-6)			1655		
LATITUDE	(DEG	MIN)	69	48.75	N
LONGITUDE	(DEG	MIN)	134	38.23	W
ICE DEPTH (m)			1.9		
FREEBOARD (m)			0.13		
SNOW DEPTH (cm)			4		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	5.784	-4.05
10-20	7.173	-3.75
20-30	7.487	-3.95
30-40	7.009	-3.84
40-50	6.969	-3.68
50-60	5.972	-3.69
60-70	5.906	-3.88
70-80	5.585	-5.22
80-90	6.519	-7.80
90-100	5.823	-9.52
100-110	5.187	-11.74
110-120	3.511	-13.70
120-130	3.291	-14.64
130-140	4.085	-15.22
140-150	3.952	-15.41
150-160	3.252	-15.62
160-170	2.814	-15.65
170-180	1.275	-15.65
180-190	1.001	-15.72
water		-18.06

STATION			PI-3		
CRUISE			9109		
DATE			26/04/91		
TIME ARRIVE (Z-6)			1424		
TIME DEPART (Z-6)			1535		
LATITUDE	(DEG	MIN)	69	54.54	N
LONGITUDE	(DEG	MIN)	134	48.54	W
ICE DEPTH (m)			2		
FREEBOARD (m)			0.13		
SNOW DEPTH (cm)			4-12		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	8.484	-3.19
10-20	6.400	-2.02
20-30	7.457	-1.81
30-40	6.537	-2.39
40-50	6.175	-2.48
50-60	6.676	-2.46
60-70	6.506	-2.36
70-80	5.521	-2.32
80-90	4.639	-2.60
90-100	5.177	-2.73
100-110	6.319	-3.35
110-120	6.956	-4.00
120-130	6.050	-5.38
130-140	6.895	-8.04
140-150	6.315	-10.88
150-160	5.827	-12.44
160-170	5.416	-12.89
170-180	5.609	-13.22
180-190	7.930	-13.77
190-200	2.986	-14.48
water		-16.46

STATION			PI-4		
CRUISE			9109		
DATE			26/04/91		
TIME ARRIVE (Z-6)			1259		
TIME DEPART (Z-6)			1410		
LATITUDE	(DEG	MIN)	69	57.63	N
LONGITUDE	(DEG	MIN)	134	52.94	W
ICE DEPTH (m)			2		
FREEBOARD (m)			0.16		
SNOW DEPTH (cm)			2		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	5.246	-5.22
10-20	4.515	-4.23
20-30	5.615	-4.29
30-40	5.525	-3.33
40-50	5.300	-2.56
50-60	5.190	-2.54
60-70	5.694	-2.32
70-80	5.362	-2.56
80-90	4.431	-2.42
90-100	4.734	-2.30
100-110	4.676	-2.10
110-120	4.516	-2.09
120-130	3.983	-2.03
130-140	4.411	-2.45
140-150	3.697	-2.67
150-160	4.622	-2.95
160-170	3.620	-3.20
170-180	3.345	-3.88
180-190	7.028	-6.08
190-200	5.636	-8.72
water		-11.90

STATION			PI-5		
CRUISE			9109		
DATE			26/04/91		
TIME ARRIVE (Z-6)			1105		
TIME DEPART (Z-6)			1240		
LATITUDE	(DEG	MIN)	70	2.28	N
LONGITUDE	(DEG	MIN)	135	1.5	W
ICE DEPTH (m)			195		
FREEBOARD (m)			0.17		
SNOW DEPTH (cm)			3		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)	
0-10	5.824	-3.63	
10-20	5.215	-2.44	
20-30	4.659	-3.11	
30-40	5.695	-2.15	
40-50	6.059		Bad-evap
50-60	6.109	-1.99	
60-70	5.992	-1.63	
70-80	5.721	-2.07	
80-90	5.131	-1.95	
90-100	5.936	-1.74	
100-110	5.532	-1.48	
110-120	4.978	-1.69	
120-130	5.389	-1.84	
130-140	4.682	-1.72	
140-150	4.732	-1.56	
150-160	4.701	-1.11	
160-170	5.003	-1.51	
170-180	6.619	-1.86	
180-187	7.375	-1.98	
187-195	9.260	-1.93	
water		-3.83	

STATION			AP-1		
CRUISE			9109		
DATE			01/05/91		
TIME ARRIVE (Z-6)			1100		
TIME DEPART (Z-6)			1200		
LATITUDE	(DEG	MIN)	69	59.74	N
LONGITUDE	(DEG	MIN)	131	27.97	W
ICE DEPTH (m)			1.7		
FREEBOARD (m)			0.1		
SNOW DEPTH (cm)			10-30		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	7.713	-4.64
10-20	7.565	-3.87
20-30	8.856	-4.49
30-40	8.021	-4.68
40-50	6.941	-4.47
50-60	4.663	-4.02
60-70	4.770	-3.88
70-80	4.575	-3.96
80-90	5.044	-4.20
90-100	5.430	-5.28
100-110	6.771	-8.11
110-120	6.610	-9.76
120-130	5.782	-11.50
130-140	6.866	-12.91
140-150	7.788	-13.44
150-160	8.192	-13.69
160-170	6.323	-14.10
water	12.156	-15.87

STATION			AP-2		
CRUISE			9109		
DATE			01/05/91		
TIME ARRIVE (Z-6)			1500		
TIME DEPART (Z-6)			1620		
LATITUDE	(DEG	MIN)	70	4.21	N
LONGITUDE	(DEG	MIN)	131	25.2	W
ICE DEPTH (m)			1.7		
FREEBOARD (m)			0.1		
SNOW DEPTH (cm)			2-10		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	8.183	-2.54
10-20	6.969	-2.26
20-30	7.114	-2.03
30-40	6.760	-2.10
40-50	5.986	-1.82
50-60	5.664	-1.87
60-70	5.320	-1.86
70-80	5.655	-2.73
80-90	5.637	-3.22
90-100	5.338	-3.47
100-110	4.591	-3.75
110-120	5.285	-3.98
120-130	5.145	-5.23
130-140	5.810	-4.32
140-150	6.945	-6.18
150-160	6.873	-9.19
160-170	6.565	-12.36
water	13.195	-15.48

STATION			AP-3		
CRUISE			9109		
DATE			01/05/91		
TIME ARRIVE (Z-6)			1320		
TIME DEPART (Z-6)			1450		
LATITUDE	(DEG	MIN)	70	8.52	N
LONGITUDE	(DEG	MIN)	131	25.16	W
ICE DEPTH (m)			1.94		
FREEBOARD (m)			0.2		
SNOW DEPTH (cm)			1-2		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	5.758	-2.93
10-20	8.833	-2.61
20-30	8.681	-2.46
30-40	7.066	-2.30
40-50	5.549	-2.05
50-60	4.969	-1.99
60-70	5.009	-1.59
70-80	7.030	-1.64
80-90	6.706	-1.37
90-100	5.756	-1.16
100-110	6.136	-1.37
110-120	5.860	-1.39
120-130	4.797	-1.25
130-140	4.665	-1.62
140-150	3.987	-2.25
150-160	3.463	-2.56
160-170	3.735	-2.98
170-180	5.281	-3.39
180-190	6.077	-3.63
190-194	10.040	-3.85
water	32.454	-7.01

STATION			AP-4		
CRUISE			9109		
DATE			01/05/91		
TIME ARRIVE (Z-6)			1217		
TIME DEPART (Z-6)			1325		
LATITUDE	(DEG	MIN)	70	15.44	N
LONGITUDE	(DEG	MIN)	131	29.24	W
ICE DEPTH (m)			1.8		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			2		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	8.314	-2.02
10-20	7.374	-1.60
20-30	8.305	-1.63
30-40	7.480	-1.55
40-50	6.258	-1.47
50-60	5.960	-1.38
60-70	5.037	-1.25
70-80	4.883	-1.15
80-90	4.882	-1.05
90-100	4.684	-0.97
100-110	5.818	-1.03
110-120	5.772	-1.08
120-130	5.737	-1.02
130-140	5.632	-1.06
140-150	6.493	-1.11
150-161	6.936	-0.86
161-170	5.005	-0.93
170-180	8.970	-0.92
water	37.117	-4.54

STATION			AP-5		
CRUISE			9109		
DATE			01/05/91		
TIME ARRIVE (Z-6)					
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	18.05	N
LONGITUDE	(DEG	MIN)	131	21.29	W
ICE DEPTH (m)			1.6		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			3		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	7.326	-2.38
10-20	7.337	-1.68
20-30	13.456	-3.08
30-40	7.807	-1.85
40-50	7.600	-1.67
50-60	7.283	-1.68
60-70	8.467	-1.59
70-80	7.526	-1.24
80-90	6.112	-1.59
90-100	8.061	-1.66
100-110	7.516	-1.46
110-120	7.763	-1.29
120-130	8.200	-1.29
130-140	6.979	-1.53
140-150	8.939	-1.84
150-160		-2.14
water	15.219	

STATION			CD-2		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)			945		
TIME DEPART (Z-6)			1045		
LATITUDE	(DEG	MIN)	70	25.01	N
LONGITUDE	(DEG	MIN)	129	39.98	W
ICE DEPTH (m)			1.82		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			5-30		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	8.259	-1.54
10-20	8.075	-0.87
20-30	8.973	-0.71
30-40	9.249	-0.88
40-50	7.360	-0.92
50-60	6.743	-0.90
60-70	5.078	-1.05
70-80	5.035	-1.06
80-90	5.768	-1.07
90-100	6.612	-1.11
100-110	8.103	-1.16
110-120	6.130	-1.22
120-130	5.134	-1.42
130-140	4.528	-1.46
140-150	5.513	-2.07
150-161	6.588	-2.46
161-170	7.141	-3.04
170-182	9.400	-5.62
water	21.137	-9.18

STATION			CD-3		
CRUISE			9109		
DATE			03/05/91		
TIME ARRIVE (Z-6)			1500		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	31.46	N
LONGITUDE	(DEG	MIN)	129	39.47	W
ICE DEPTH (m)			1.62		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)					

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	9.747	-1.78
10-20	6.692	-1.19
20-30	6.728	-1.04
30-40	7.755	-1.09
40-50	5.221	-0.89
50-60	4.503	-0.81
60-70	4.938	-0.71
70-80	5.57	-0.35
80-90	4.965	-0.81
90-100	4.235	-0.92
100-110	3.731	-1.17
110-120	5.228	-1.46
120-130	4.522	-1.29
130-140	3.783	-1.15
140-150	4.184	-1.30
150-162	6.73	-1.87
water	33.032	-3.02

STATION			CD-4		
CRUISE			9109		
DATE			04/05/91		
TIME ARRIVE (Z-6)			1300		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	39.95	N
LONGITUDE	(DEG	MIN)	129	40.83	W
ICE DEPTH (m)			1.96		
FREEBOARD (m)			0.18		
SNOW DEPTH (cm)			4-13		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	6.109	-1.03
10-20	5.776	-1.38
20-30	6.248	-1.41
30-40	6.179	-1.42
40-50	6.145	-1.46
50-60	5.698	-1.41
60-70	5.725	-1.46
70-80	5.205	-1.33
80-90	5.121	-0.85
90-100	5.3	-0.64
100-110	5.502	-0.58
110-120	6.263	-0.68
120-130	6.527	-0.90
130-140	5.034	-0.81
140-150	4.491	-0.84
150-160	5.024	-0.98
160-170	4.422	-1.01
170-180	5.259	-0.80
180-188	5.714	-0.73
188-196	10.372	-1.36
water	33.177	-3.26

STATION			CD-5		
CRUISE			9109		
DATE			04/05/91		
TIME ARRIVE (Z-6)			1200		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	48.32	N
LONGITUDE	(DEG	MIN)	129	40.35	W
ICE DEPTH (m)			0.97		
FREEBOARD (m)			0.08		
SNOW DEPTH (cm)			1-5		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	10.221	-1.78
10-20	8.589	-1.37
20-30	6.915	-1.46
30-40	6.19	-1.07
40-50	5.898	-0.85
50-60	4.937	-0.66
60-70	4.682	-0.65
70-80	4.627	-0.73
80-88	4.671	-0.59
88-97	6.728	-0.46
water	32.073	-4.32

STATION			CM-1		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)			1845		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	45.01	N
LONGITUDE	(DEG	MIN)	133	18	W
ICE DEPTH (m)			1.86		
FREEBOARD (m)			0.1		
SNOW DEPTH (cm)					

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	5.929	-4.57
10-20	6.588	-4.39
20-30	5.602	-5.52
30-40	4.504	-6.15
40-50	4.263	-7.69
50-60	3.918	-9.30
60-70	3.571	-10.36
70-80	2.453	-12.34
60-90	3.281	-12.72
90-100	3.847	-13.42
100-110	3.986	-14.01
110-120	2.731	-14.44
120-130	2.393	-14.80
130-140	1.227	-14.73
140-150	2.138	-14.83
150-160	2.491	-14.69
160-170	2.894	-14.65
170-180	2.241	
180-186	1.132	-14.74
water	2.578	-17.72

STATION			CM-2		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)			805		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	50.05	N
LONGITUDE	(DEG	MIN)	133	21.32	W
ICE DEPTH (m)			1.86		
FREEBOARD (m)			0.13		
SNOW DEPTH (cm)			2-6		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	2.736	-4.31
10-20	6.678	-3.81
20-30	8.260	-3.82
30-40	7.333	-3.61
40-50	5.621	-3.73
50-60	4.543	-4.51
60-70	3.959	-5.69
70-80	3.523	-6.78
80-90	3.379	-8.51
90-100	4.057	-9.90
100-110	3.876	-10.93
110-120	4.187	-11.76
120-130	4.394	-12.59
130-140	4.366	-13.45
140-150	5.218	-13.66
150-160	4.461	-13.99
160-170	4.806	-14.16
170-180	3.829	-14.24
180-186	3.818	-13.95
water	6.171	-16.60

STATION			CM-3		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)			1707		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	56.15	N
LONGITUDE	(DEG	MIN)	133	21.57	W
ICE DEPTH (m)			1.65		
FREEBOARD (m)			0.09		
SNOW DEPTH (cm)			5-9		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	5.368	-2.42
10-20	6.694	-2.15
20-30	5.166	-1.88
30-40	6.683	-1.73
40-50	6.365	-1.96
50-60	5.711	-1.87
60-70	5.804	-1.67
70-80	6.198	-1.58
80-90	6.268	-1.59
90-100	6.379	-1.57
100-110	5.226	-2.01
110-120	4.432	-2.43
120-130	4.348	-2.60
130-140	6.878	-3.79
140-150	6.269	-5.04
150-160	5.396	-5.67
160-165	7.700	-6.93
water	21.991	-10.06

STATION			CM-4		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)			1600		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	0.99	N
LONGITUDE	(DEG	MIN)	133	24.05	W
ICE DEPTH (m)			1.93		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			2-10		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	5.008	-3.51
10-20	5.236	-2.25
20-30	5.732	-3.17
30-40	6.098	-2.01
40-50	6.261	-1.80
50-60	5.991	-1.65
60-70	5.774	-0.86
70-80	5.414	-1.13
80-90	4.571	-1.60
90-100	6.301	-1.41
100-110	6.524	-1.29
110-120	6.869	-1.50
120-130	6.866	-1.51
130-140	6.725	-1.23
140-150	3.998	-1.27
150-160	4.421	-1.40
160-170	4.697	-1.14
170-180	6.308	-1.46
180-193	7.844	-1.41
water	32.029	-3.43

STATION			L-2		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)			1707		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	54.03	N
LONGITUDE	(DEG	MIN)	132	13.88	W
ICE DEPTH (m)			2		
FREEBOARD (m)			0.17		
SNOW DEPTH (cm)			1-2		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-12	2.934	-3.60
12-24	8.239	-2.92
24-30	10.128	-3.32
30-40	8.559	-2.79
40-50	7.782	-2.98
50-60	6.367	-3.12
60-70	5.068	-2.69
70-80	4.970	-2.78
80-90	5.931	-2.98
90-100	5.417	-3.47
100-110	4.447	-4.16
110-120	5.166	-4.96
120-130	5.003	-5.64
130-140	3.940	-6.88
140-150	5.586	-9.34
150-160	5.609	-11.14
160-170	5.793	-12.21
170-180	5.678	-13.20
180-190	5.360	-14.12
190-200	7.100	-14.21
water	4.025	-16.80

STATION			L-3		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)			1600		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	16.06	N
LONGITUDE	(DEG	MIN)	130	31.69	W
ICE DEPTH (m)			1.87		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			2-3		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	3.686	-2.82
10-20	8.236	-2.47
20-30	8.315	-2.68
30-40	9.118	-2.18
40-50	7.387	-2.32
50-60	6.148	-2.37
60-70	4.864	-1.73
70-80	5.126	-1.60
80-90	5.507	-1.69
90-100	5.426	-1.54
100-110	7.419	-1.69
110-120	5.340	-1.57
120-130	6.256	-2.21
130-140	4.588	-2.80
140-150	3.339	-2.93
150-160	5.171	-4.12
160-170	5.338	-7.06
170-180	3.846	-11.25
180-187	5.250	-12.28
water	8.263	-14.83

STATION			MB-1		
CRUISE			9109		
DATE			09/05/91		
TIME ARRIVE (Z-6)			1430		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	33.18	N
LONGITUDE	(DEG	MIN)	134	4.51	W
ICE DEPTH (m)			2		
FREEBOARD (m)			0.13		
SNOW DEPTH (cm)			2-8		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10		-15.58
10-20		-8.98
20-30		-8.96
30-40		-8.69
40-50		-8.68
50-60		-8.57
60-70		-8.35
70-80		-10.15
80-90		-12.06
90-100		-12.07
100-110		-13.83
110-120		-14.30
120-130		-14.36
130-140		-14.48
140-150		-15.42
150-160		-15.78
160-170		-14.64
170-180		-15.44
180-190		-15.61
190-200		-16.26
water		-18.59

STATION			KP-2		
CRUISE			9109		
DATE			11/05/91		
TIME ARRIVE (Z-6)			1550		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	23.94	N
LONGITUDE	(DEG	MIN)	138	7.47	W
ICE DEPTH (m)			1.88		
FREEBOARD (m)			0.18		
SNOW DEPTH (cm)			5-8		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	2.5	-4.36
10-20	5.1	-1.63
20-30	4.9	-1.50
30-37	5.8	-1.16
37-50	6.1	-1.23
50-60	6.2	-1.29
60-70	6	-1.26
70-80	5.7	-1.73
80-90	5.4	-1.68
90-100	5.2	-1.32
100-110	4.9	-1.36
110-120	4.4	-1.38
120-130	4.3	-1.05
130-140	5.8	-1.73
140-150	4.1	-2.06
150-160	3	-5.88
160-170	3.4	-8.97
170-180	4	-11.08
180-188	4.1	-11.97
water	10.4	-14.76

STATION			HI-1		
CRUISE			9109		
DATE			11/05/91		
TIME ARRIVE (Z-6)			1250		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	28	N
LONGITUDE	(DEG	MIN)	138	50.58	W
ICE DEPTH (m)			1.88		
FREEBOARD (m)			0.05		
SNOW DEPTH (cm)			30		

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	8	-2.07
10-20	9.6	-1.73
20-30	7.9	-1.79
30-40	5.9	-2.32
40-50	6.7	-2.08
50-60	6.7	-2.06
60-70	6.7	-1.96
70-80	5.4	-2.06
80-90	5.9	-2.13
90-100	5	-1.85
100-110	5.5	-1.91
110-120	5.5	-1.78
120-130	5.3	-1.60
130-140	5.6	-1.44
140-150	5.6	-2.08
150-160	4.1	-4.21
160-170	3.6	-8.54
170-180	3.3	-10.24
180-188	3.9	-11.55
water	10.8	-14.75

STATION			HI-2		
CRUISE			9109		
DATE			11/05/91		
TIME ARRIVE (Z-6)			1350		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	36.82	N
LONGITUDE	(DEG	MIN)	138	47.84	W
ICE DEPTH (m)			1.89		
FREEBOARD (m)			0.17		
SNOW DEPTH (cm)					

CORE INTERVAL (cm)	SALINITY (psu)	δO-18 (vs V-SMOW)
0-10	2.2	-12.18
10-20	3.9	-9.18
20-30	5.8	-2.82
30-40	6.1	-1.96
40-50	6.1	-1.64
50-60	6.5	-1.97
60-70	6	-1.89
70-80	5.4	-1.74
80-90	5.1	-1.48
90-100	4.8	-1.19
100-110	4.7	-1.16
110-120	4.9	-1.06
120-130	5.4	-1.07
130-140	5.1	-1.13
140-150	4.7	-1.15
150-160	5.3	-0.98
160-170	5.1	-0.93
170-180	5.8	-0.81
180-189	9.3	-1.46
water	31	-3.89

STATION	GI-1
CRUISE	9109
DATE	11/05/91
TIME ARRIVE (Z-6)	1120
TIME DEPART (Z-6)	
LATITUDE (DEG MIN)	69 30.24 N
LONGITUDE (DEG MIN)	136 9.49 W
ICE DEPTH (m)	1.91
FREEBOARD (m)	0.13
SNOW DEPTH (cm)	3-30

CORE INTERVAL (cm)	SALINITY (psu)	δO–18 (vs V–SMOW)
0-10	3.9	-9.63
10-20	2.3	-13.28
20-30	2.1	-14.84
30-40	0.4	-16.01
40-50	0.1	-16.03
50-60	0.2	-16.12
60-70	0.1	-16.58
70-80	0	-16.45
80-90	0.3	-16.26
90-100	0	-16.02
100-110	0.1	-16.10
110-120	0.2	-16.17
120-130	0.2	-16.53
130-140	0.1	-16.34
140-150	0.2	-16.11
150-160	0.1	-16.17
160-170	0.2	-16.11
170-180	0.3	-16.03
180-191	0.3	-15.80
water	0.5	-18.24

STATION	GI-2
CRUISE	9109
DATE	11/05/91
TIME ARRIVE (Z-6)	1740
TIME DEPART (Z-6)	
LATITUDE (DEG MIN)	69 38.05 N
LONGITUDE (DEG MIN)	136 21.95 W
ICE DEPTH (m)	1.84
FREEBOARD (m)	0.12
SNOW DEPTH (cm)	7

CORE INTERVAL (cm)	SALINITY (psu)	δO–18 (vs V–SMOW)
0-10	1	-8.75
10-20	4.7	-5.49
20-30	6	-4.75
30-40	6.8	-4.71
40-50	6.6	-4.57
50-60	5.7	-4.27
60-70	5.2	-4.29
70-80	4.8	-6.35
80-90	5	-8.31
90-100	3.8	-11.31
100-110	3.2	-13.55
110-120	4.3	-14.03
120-130	4	-14.58
130-140	3	-15.14
140-150	3.1	-15.51
150-160	3.7	-15.44
160-170	3.6	-15.40
170-184	3.3	-15.26
water	2.1	-18.38

STATION			PI-1		
CRUISE			9109		
DATE			30/04/91		
TIME ARRIVE (Z-6)			1340		
TIME DEPART (Z-6)			1510		
LATITUDE	(DEG	MIN)	69	43.84	N
LONGITUDE	(DEG	MIN)	134	31.25	W
ICE DEPTH (m)			2.02		
FREEBOARD (m)			0.13		
SNOW DEPTH (cm)			9-20		
UNDER ICE WATER DEPTH (m)			1.98		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	1.195	-18.21
1.5	1.626	-17.83

STATION			PI-2		
CRUISE			9109		
DATE			30/04/91		
TIME ARRIVE (Z-6)			1100		
TIME DEPART (Z-6)			1330		
LATITUDE	(DEG	MIN)	69	48.75	N
LONGITUDE	(DEG	MIN)	134	38.22	W
ICE DEPTH (m)			1.9		
FREEBOARD (m)			0.12		
SNOW DEPTH (cm)			3-12		
UNDER ICE WATER DEPTH (m)			5.6		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	2.577	-18.04
2	1.373	-17.69
4	11.282	-13.99

STATION			PI-3		
CRUISE			9109		
DATE			29/04/91		
TIME ARRIVE (Z-6)			1005		
TIME DEPART (Z-6)			1200		
LATITUDE	(DEG	MIN)	69	55.04	N
LONGITUDE	(DEG	MIN)	134	47.96	W
ICE DEPTH (m)			1.93		
FREEBOARD (m)			0.25		
SNOW DEPTH (cm)			9-18		
UNDER ICE WATER DEPTH (m)			12.07		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	5.701	-16.58
2	6.868	-16.22
4	11.621	-14.50
7	30.792	-4.87
10	31.253	-4.39
13.5	31.656	-4.16

STATION			PI-4		
CRUISE			9109		
DATE			29/04/91		
TIME ARRIVE (Z-6)			1210		
TIME DEPART (Z-6)			1418		
LATITUDE	(DEG	MIN)	69	57.92	N
LONGITUDE	(DEG	MIN)	133	58.72	W
ICE DEPTH (m)			1.9		
FREEBOARD (m)			0.2		
SNOW DEPTH (cm)			5-9		
UNDER ICE WATER DEPTH (m)			17.4		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	16.751	-12.52
2	17.685	-11.96
4	19.086	-11.00
7	30.810	-4.30
10	31.802	-4.07
15	31.899	-3.91

STATION			PI-5		
CRUISE			9109		
DATE			26/04/91		
TIME ARRIVE (Z-6)			1156		
TIME DEPART (Z-6)			1605		
LATITUDE	(DEG	MIN)	70	2.02	N
LONGITUDE	(DEG	MIN)	135	0	W
ICE DEPTH (m)			2.00		
FREEBOARD (m)			0.20		
SNOW DEPTH (cm)			2-15		
UNDER ICE WATER DEPTH (m)			27.00		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	31.871	-3.89
2	31.862	-3.99
4	31.873	-4.10
7	31.889	-4.34
10	31.894	-4.12
15	31.959	-3.96
20	32.174	-3.83
23	32.208	-3.67
25	32.221	-3.70

STATION			L-1		
CRUISE			9109		
DATE			30/04/91		
TIME ARRIVE (Z-6)			1523		
TIME DEPART (Z-6)			1652		
LATITUDE	(DEG	MIN)	69	53	N
LONGITUDE	(DEG	MIN)	133	59.98	W
ICE DEPTH (m)			1.77		
FREEBOARD (m)			0.21		
SNOW DEPTH (cm)			5-12		
UNDER ICE WATER DEPTH (m)			7.23		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	4.897	-17.13
2	5.472	-16.92
4	10.710	-14.46
7	29.670	-5.43

STATION			L-2		
CRUISE			9109		
DATE			02/05/91		
TIME ARRIVE (Z-6)			1953		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	54.04	N
LONGITUDE	(DEG	MIN)	132	13.84	W
ICE DEPTH (m)			1.98		
FREEBOARD (m)			0.17		
SNOW DEPTH (cm)			2		
UNDER ICE WATER DEPTH (m)			9.32		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	4.126	-17.35
2	4.205	-17.19
4	13.929	-13.39
5	29.320	-5.80
7	30.790	-5.01

STATION			L-3		
CRUISE			9109		
DATE			07/05/91		
TIME ARRIVE (Z-6)			1047		
TIME DEPART (Z-6)			1222		
LATITUDE	(DEG	MIN)	70	16.14	N
LONGITUDE	(DEG	MIN)	130	29.65	W
ICE DEPTH (m)			1.9		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			5-20		
UNDER ICE WATER DEPTH (m)			6.6		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	8.314	-15.21
2	8.381	-15.16
4	30.402	-4.81
6	30.726	-4.65

STATION	CM-1
CRUISE	9109
DATE	30/04/91
TIME ARRIVE (Z-6)	1715
TIME DEPART (Z-6)	
LATITUDE (DEG MIN)	69 45.01 N
LONGITUDE (DEG MIN)	133 18 W
ICE DEPTH (m)	1.83
FREEBOARD (m)	0.1
SNOW DEPTH (cm)	
UNDER ICE WATER DEPTH (m)	6.17

SAMPLE DEPTH (m)	SALINITY psu	δO−18 (vs V−SMOW)
0	2.829	-17.55
2	3.812	-17.52
4	9.185	-15.49
6	28.676	-6.17

STATION	CM-2
CRUISE	9109
DATE	28/04/91
TIME ARRIVE (Z-6)	1743
TIME DEPART (Z-6)	
LATITUDE (DEG MIN)	69 50 N
LONGITUDE (DEG MIN)	133 19.99 W
ICE DEPTH (m)	1.8
FREEBOARD (m)	0.13
SNOW DEPTH (cm)	2-6
UNDER ICE WATER DEPTH (m)	10.7

SAMPLE DEPTH (m)	SALINITY psu	δO−18 (vs V−SMOW)
0	6.499	-16.50
2	6.621	-16.58
4	9.171	-15.62
7	30.417	-5.49
10	30.616	-5.27

STATION			CM-3		
CRUISE			9109		
DATE			28/04/91		
TIME ARRIVE (Z-6)			1406		
TIME DEPART (Z-6)			1730		
LATITUDE	(DEG	MIN)	69	56.14	N
LONGITUDE	(DEG	MIN)	133	21.61	W
ICE DEPTH (m)			1.6		
FREEBOARD (m)			0.09		
SNOW DEPTH (cm)			5-9		
UNDER ICE WATER DEPTH (m)			20.9		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	23.289	-8.83
2	23.931	-8.87
4	31.537	-4.54
7	31.903	-4.04
10	31.953	-4.15
15	32.052	-3.97
20	32.181	-3.81

STATION			CM-4		
CRUISE			9109		
DATE			28/04/91		
TIME ARRIVE (Z-6)			950		
TIME DEPART (Z-6)			1355		
LATITUDE	(DEG	MIN)	70	0.98	N
LONGITUDE	(DEG	MIN)	133	23.95	W
ICE DEPTH (m)			2		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			2-10		
UNDER ICE WATER DEPTH (m)			28		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	32.102	-4.26
2	32.089	-3.64
4	32.130	-4.04
7	32.156	-4.01
10	32.165	-3.86
15	32.307	-3.45
20	32.172	-3.76
25	32.222	-3.73

STATION			CM-5		
CRUISE			9109		
DATE			26/04/91		
TIME ARRIVE (Z-6)			1623		
TIME DEPART (Z-6)			2013		
LATITUDE	(DEG	MIN)	70	8.06	N
LONGITUDE	(DEG	MIN)	133	25.23	W
ICE DEPTH (m)			1.9		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			3		
UNDER ICE WATER DEPTH (m)			36.6		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	32.242	-3.42
2	32.239	-3.33
4	32.240	-3.36
7	32.241	-3.29
10	32.241	-3.24
15	32.251	-3.14
20	32.270	-3.17
30	32.312	-3.46
35	32.326	-3.59

STATION			C-6		
CRUISE			9109		
DATE			29/04/91		
TIME ARRIVE (Z-6)			1445		
TIME DEPART (Z-6)			1629		
LATITUDE	(DEG	MIN)	70	28.29	N
LONGITUDE	(DEG	MIN)	133	51.47	W
ICE DEPTH (m)			1.1		
FREEBOARD (m)			0.1		
SNOW DEPTH (cm)			15		
UNDER ICE WATER DEPTH (m)			59.15		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	31.985	-3.29
2	31.955	-3.06
4	31.954	-3.17
7	31.956	-3.26
10	31.958	-3.30
15	31.965	-3.55
20	31.976	-3.06
30	31.989	-3.40
40	32.117	-3.14

STATION	AP-1
CRUISE	9109
DATE	07/05/91
TIME ARRIVE (Z-6)	1446
TIME DEPART (Z-6)	1612

LATITUDE	(DEG	MIN)	69	59.74	N
LONGITUDE	(DEG	MIN)	131	27.96	W

ICE DEPTH (m)	1.7
FREEBOARD (m)	0.1
SNOW DEPTH (cm)	10-30
UNDER ICE WATER DEPTH (m)	9.8

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	4.089	-17.11
2	4.054	-17.06
4	4.117	-16.93
7	30.682	-4.57
9	30.799	-4.52

STATION	AP-2
CRUISE	9109
DATE	07/05/91
TIME ARRIVE (Z-6)	1238
TIME DEPART (Z-6)	1434

LATITUDE	(DEG	MIN)	70	4.22	N
LONGITUDE	(DEG	MIN)	131	25.19	W

ICE DEPTH (m)	1.6
FREEBOARD (m)	0.1
SNOW DEPTH (cm)	2-10
UNDER ICE WATER DEPTH (m)	12.4

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	8.246	-15.43
2	8.333	-15.90
4	11.651	-14.31
7	31.475	-4.65
10	31.798	-4.43

STATION	AP-3
CRUISE	9109
DATE	29/04/91
TIME ARRIVE (Z-6)	1859
TIME DEPART (Z-6)	2000

LATITUDE	(DEG	MIN)	70	8.52	N
LONGITUDE	(DEG	MIN)	131	25.13	W

ICE DEPTH (m)	1.9
FREEBOARD (m)	0.2
SNOW DEPTH (cm)	1-2
UNDER ICE WATER DEPTH (m)	17.1

SAMPLE DEPTH (m)	SALINITY psu	δO−18 (vs V−SMOW)
0	29.787	-5.98
2	29.851	-5.96
4	30.085	-5.59
7	31.806	-4.23
10	32.177	-4.61
15	32.362	-4.24

STATION	AP-4
CRUISE	9109
DATE	29/04/91
TIME ARRIVE (Z-6)	1710
TIME DEPART (Z-6)	1840

LATITUDE	(DEG	MIN)	70	15.45	N
LONGITUDE	(DEG	MIN)	131	28.5	W

ICE DEPTH (m)	1.9
FREEBOARD (m)	0.15
SNOW DEPTH (cm)	1-6
UNDER ICE WATER DEPTH (m)	21.1

SAMPLE DEPTH (m)	SALINITY psu	δO−18 (vs V−SMOW)
0	32.510	-3.31
2	32.478	-3.25
4	32.480	-3.18
7	32.479	-3.23
10	32.478	-3.17
15	32.480	-3.48
20	32.482	-3.21

STATION			AP-5		
CRUISE			9109		
DATE			26/04/91		
TIME ARRIVE (Z-6)			2045		
TIME DEPART (Z-6)			2200		
LATITUDE	(DEG	MIN)	70	18.03	N
LONGITUDE	(DEG	MIN)	131	21.33	W
ICE DEPTH (m)			1.50		
FREEBOARD (m)			0.15		
SNOW DEPTH (cm)			3		
UNDER ICE WATER DEPTH (m)			25.00		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	32.383	-3.24
2	32.384	-3.28
4	32.382	-3.49
7	32.387	-3.36
10	32.381	-3.26
15	32.381	-3.11
20	32.382	-3.61

STATION			WESTC		
CRUISE			9109		
DATE			30/04/91		
TIME ARRIVE (Z-6)			1800		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	21.4	N
LONGITUDE	(DEG	MIN)	133	57.67	W
ICE DEPTH (m)			1.9		
FREEBOARD (m)			0.16		
SNOW DEPTH (cm)					
UNDER ICE WATER DEPTH (m)			7.3		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
2		-18.13

STATION			CB-4		
CRUISE			9109		
DATE			06/05/91		
TIME ARRIVE (Z-6)					
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	29.75	N
LONGITUDE	(DEG	MIN)	128	28.4	W
ICE DEPTH (m)			1.92		
FREEBOARD (m)			0.13		
SNOW DEPTH (cm)					
UNDER ICE WATER DEPTH (m)			10.08		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	33.118	-4.66
4	33.526	-4.88
8	34.608	-4.42

STATION			CB-7B		
CRUISE			9109		
DATE			07/05/91		
TIME ARRIVE (Z-6)					
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	38.88	N
LONGITUDE	(DEG	MIN)	128	16.21	W
ICE DEPTH (m)			2.08		
FREEBOARD (m)			0.2		
SNOW DEPTH (cm)					
UNDER ICE WATER DEPTH (m)			14		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	32.508	-3.64
7	33.118	-3.72
13	33.280	-3.83

STATION			LB-1		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)					
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	70	15.03	N
LONGITUDE	(DEG	MIN)	128	52.4	W
ICE DEPTH (m)			2		
FREEBOARD (m)			0.17		
SNOW DEPTH (cm)			1-10		
UNDER ICE WATER DEPTH (m)			10.5		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
2	31.424	-5.46
9	34.318	-4.86

STATION			REV-2		
CRUISE			9109		
DATE			05/05/91		
TIME ARRIVE (Z-6)			1530		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	56	N
LONGITUDE	(DEG	MIN)	133	20	W
ICE DEPTH (m)			1.7		
FREEBOARD (m)			0.13		
SNOW DEPTH (cm)			3-4		
UNDER ICE WATER DEPTH (m)			20.4		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	22.156	-9.30
2	23.494	-8.76
4	30.075	-4.88
7	31.814	-4.30
10	31.908	-4.34
15	31.973	-4.16
19	32.133	-3.85

STATION			ISZ-1		
CRUISE			9109		
DATE			03/05/91		
TIME ARRIVE (Z-6)			1530		
TIME DEPART (Z-6)			1700		
LATITUDE	(DEG	MIN)	70	11.6	N
LONGITUDE	(DEG	MIN)	133	39.73	W
ICE DEPTH (m)			0		
FREEBOARD (m)					
SNOW DEPTH (cm)					
WATER DEPTH (m)			43	**	

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	31.954	-3.19
2	32.015	-3.62
4	32.019	-3.60
7	31.994	-3.47
10	32.089	-3.54
15	32.092	-3.44
20	32.104	-3.43
30	32.158	-3.35
40	32.296	-3.25

** sampled from an open lead

STATION			MB		
CRUISE			9109		
DATE			09/05/91		
TIME ARRIVE (Z-6)			1354		
TIME DEPART (Z-6)					
LATITUDE	(DEG	MIN)	69	33.18	N
LONGITUDE	(DEG	MIN)	134	4.51	W
ICE DEPTH (m)			1.95		
FREEBOARD (m)			0.1		
SNOW DEPTH (cm)					
UNDER ICE WATER DEPTH (m)			28.65		

SAMPLE DEPTH (m)	SALINITY psu	δO-18 (vs V-SMOW)
0	3.476	-18.74
4	19.403	-9.87
7	19.910	-9.59
10	21.702	-8.14
15	21.664	-8.44
29	21.730	-8.50

5 APPENDIX B, SOFTWARE CONTROL PROGRAMS

This appendix contains the following two programs written by David Macdonald:

Massmenu.bas The program to control the mass-spec operation.

oxy.bas The program to control the equilibrator.

```
'        ******************************
'        *  PROGRAM MASSMENU.BAS     *
'        ******************************

'           - written by D.Macdonald  --> Dec 1991
'           - adapted from program 'GIRDx'
'           - uses Quinn-Curtis SCIENCE & ENGINEERING TOOLS
'                (esp. for graphics screen dump)
'           qb.exe path + massmenu.mak where the '.mak' file
'           has the following contents:
'                   MASSMENU.BAS
'                   C:\SETOOLS\HPPLOT.BAS
'                   C:\SETOOLS\SEGRAPH.BAS
'                   C:\SETOOLS\WORLDDR.BAS

'           - * when compiling, use "Produce debug code" option
'             for proper operation under DV (who knows why??)

'           ***** SUBROUTINES ARE IN ALPHABETICAL ORDER *****
'           *****      FOLLOWING MAIN ROUTINE         *****

DECLARE SUB ClearWindow ()
DECLARE SUB ScreenDump (printer%, Prnport%, res%, xm%, ym%, rv%, ff%, prnerr%)
DECLARE SUB DefGraphWindow (xx1%, yy1%, xx2%, yy2%, win%)
DECLARE SUB InitSEGraphics ()
DECLARE SUB SetCurrentWindow (win%)
DECLARE SUB CRTGraphOn ()


    COMMON maxc%, maxr%, maxv%, FALSE%, TRUE%
    maxc% = 10: maxr% = 20: maxv% = 256: FALSE% = 0: TRUE% = -1

    DEFINT A-Z
    DIM X1 AS INTEGER, y1 AS INTEGER, X2 AS INTEGER
    DIM y2 AS INTEGER, win AS INTEGER
    DIM chauvenet$(25)
    DIM IR.READ.1#(2, 25, 25), IR.READ.2#(2, 25, 25)
    DIM RATIO.1#(2, 25), RATIO.2#(2, 25), DEL#(2), DEVIATION#(2)
    DIM GAS#(2, 25), TOTAL.1#(2, 25), TOTAL.2#(2, 25), DEL1#(25), DEL2#(25)
    DIM DEL.1(2, 25, 25), DEL.2(2, 25, 25), GAS.DEV#(2), GAS.MEAN#(2)
    DIM SD.1#(2, 25), SD.2#(2, 25), REJECT.1(2, 25), REJECT.2(2, 25)
    DIM MEAN.1#(2, 25), MEAN.2#(2, 25), CYCLES.DEL.1(25), CYCLES.DEL.2(25)
    DIM FINAL.MEAN.1#(2), FINAL.MEAN.2#(2)
    DIM FINAL.DEV.1#(2), FINAL.DEV.2#(2)
    DIM DEL.ERROR#(2)
    DIM item$(11)
    X1% = 0: y1% = 0: X2% = 639: y2% = 479: win% = 1


GOSUB initialize
```

```
'...................... START OF MAIN ......................................
main:
    CRTGraphOn
    InitSEGraphics
    CALL DefGraphWindow(X1%, y1%, X2%, y2%, win%)
    CALL DefGraphWindow(0, 41, 639, 479, 2)
    CALL SetCurrentWindow(1)
    GOSUB get.cfg.list
    GOSUB header
    FOR i = 1 TO 11: READ item$(i): NEXT i
    DATA Edit setup,Load/delete setup,Analyse gas,Valve select,Retrieve datafile
    DATA Recalc datafile,Plot hardcopy,Data hardcopy,Printer Form Feed,Printer config,Quit
    menu = 1: item = 1
    FOR i = 1 TO 9
        READ lower.limit!(i), upper.limit!(i)
    NEXT i
    DATA 1,6,3,25,1,2.32,3,25,1,2.32,0,60,0,0,0,500,0,0
    FOR i = 1 TO 23
        READ chauvenet$(i + 2)
    NEXT i
    DATA 1.37,1.53,1.64,1.73,1.80,1.86,1.92,1.96,2.00,2.04,2.07,2.10,2.13
    DATA 2.15,2.18,2.20,2.22,2.24,2.26,2.28,2.29,2.31,2.32
    GOSUB footer
    WHILE item <> 11
        GOSUB menu1: maxitem = 11: GOSUB selection
        SELECT CASE item
            CASE 1
                GOSUB edit.setup
            CASE 2
                GOSUB load.setup
            CASE 3
                GOSUB analyse.gas
            CASE 4
                GOSUB valve.select
            CASE 5
                GOSUB retrieve.datafile
            CASE 6
                GOSUB recalc.datafile
            CASE 7
                IF NO.DATA = FALSE THEN
                    GOSUB plot.hardcopy
                ELSE
                    BEEP
                END IF
            CASE 8
                IF NO.DATA = FALSE THEN
                    GOSUB data.hardcopy
                ELSE
                    BEEP
                END IF
            CASE 9
                GOSUB form.feed
```

```
            CASE 10
                  GOSUB printer.config
      END SELECT
   WEND
   SCREEN 0
   CLS
   END
'..................... END OF MAIN ........................................


'............ SUBROUTINES LISTED ALPHABETICALLY ...........................

abort:
   abort.flag = TRUE
   RETURN

analyse.gas:
   IF setup.edited THEN
      BEEP
      LOCATE 15, 40: PRINT "Setup file not saved! Press any key...";
      a$ = INPUT$(1)
      LOCATE 15, 40: PRINT SPC(39);
      RETURN
   END IF
   'LOCATE 26, 1: PRINT "we are in analyse gas": a$ = INPUT$(1): RETURN
   IF STARTUP THEN
      GOSUB init.IEEE.488
      GOSUB select.valve1
      IR.OUT$ = "V2"
      GOSUB IR.output.routine
      STARTUP = FALSE
   END IF
   SCALE.CALCULATED = FALSE  'WE HAVE NOT CALCULATED THE SCALE FOR
GRAPH
   GOSUB blank.menu
   CALL SetCurrentWindow(1)
   ClearWindow
new.title:
   LOCATE 10, 10
   INPUT "Enter Title For Run ", title$      'ENTER TITLE OF RUN
   IF LEN(title$) > 8 THEN
       LOCATE 15, 10
       PRINT "*** Title Must Be At Most 8 Characters ***";
       BEEP
       a$ = INPUT$(1)
       LOCATE 10, 10: PRINT SPC(30);
       LOCATE 15, 10: PRINT SPC(50);
       GOTO new.title
   END IF

   abort.flag = FALSE
   ON KEY(1) GOSUB abort
```

```
KEY(1) ON

LOCATE 10, 10: PRINT SPC(30);
t$ = TIME$              'GET STARTING TIME AND DATE
D$ = DATE$
GOSUB prepare.graphics.screen
IF AUTO.HARDCOPY THEN GOSUB print.header
LINE.PLOT = FALSE       'SET PLOTTING FLAG FALSE
'
' FIRST WE OBTAIN THE READINGS OF OUR TWO NUMERATORS AND
' THEN TAKE A RATIO PAIR OF READING, AND THEN DISPLAY THE
' DENOMINATOR VALUES FOR OUR TWO GASES.
'
GAS.MEAN#(1) = 0!
GAS.MEAN#(2) = 0!

'start CYCLE
FOR CYCLE = 1 TO CYCLES
'   cursor.vert.position = CSRLIN
'   VIEW PRINT
'   LOCATE 3, 12: PRINT CYCLE;
'   LOCATE 4, 12: PRINT "1";
'   LOCATE 5, 12: PRINT "1";
'   VIEW PRINT 19 TO 29
'   LOCATE cursor.vert.position, 2
 CYCLES.DEL.1(CYCLE) = FALSE    'SET CYCLE DELETION ARRAYS FALSE
 CYCLES.DEL.2(CYCLE) = FALSE
'start GAS
 FOR GAS = 1 TO 2
  IR.OUT$ = "M1T100"  'SET UP RATIO MODE AND APPROX. 1 SEC INT.
  GOSUB IR.output.routine       ' OUTPUT TO IR
  IF GAS = 1 THEN GOSUB select.valve1 ELSE GOSUB select.valve2
  GOSUB HALT       ' PAUSE FOR GAS TO ENTER
  FOR READING = 1 TO TOTAL.INTEGRATIONS
   IF abort.flag THEN
      VIEW PRINT
      NO.DATA = FALSE
      GOSUB select.valve1
      ClearWindow
      GOSUB header
      KEY(1) OFF
      IF INSTR(valve$, "2") <> 0 THEN GOSUB select.valve1
      RETURN
   END IF
   GOSUB trigger.and.read.IR   'TRIGGER AND READ THE IR
   GOSUB plot.data 'plot data
   IR.READ.1#(GAS, CYCLE, READING) = NUMERATOR.1# 'STORE READINGS
   IR.READ.2#(GAS, CYCLE, READING) = NUMERATOR.2#
   DEL.1(GAS, CYCLE, READING) = FALSE  'SET DELETED READING FALSE
   DEL.2(GAS, CYCLE, READING) = FALSE
'   cursor.vert.position = CSRLIN
'   VIEW PRINT
```

```
'    LOCATE 4, 12: PRINT GAS;
'    LOCATE 5, 12: PRINT READING;
'    VIEW PRINT 19 TO 29
'    LOCATE cursor.vert.position, 2
     NEXT READING
     IR.OUT$ = "M0T1000"    ' SET TO DVM 1 SEC INT.
     GOSUB IR.output.routine         'OUTPUT TO IR
     GOSUB trigger.and.read.IR        ' READ IN READING
     a# = IR.READ.1#(GAS, CYCLE, TOTAL.INTEGRATIONS) / 10!'CALC MASS 44
     GAS#(GAS, CYCLE) = (NUMERATOR.1# / a#) * RANGE
     GAS.MEAN#(GAS) = GAS.MEAN#(GAS) + GAS#(GAS, CYCLE)' sum for mean
     CHANGE = FALSE       'SET READING DELEATED FLAG FALSE
     SECOND = FLASE       'SET SECOND STATISTICS CALCULATED FALSE
newcalc:
     GOSUB calc.per.cycle.and.gas
     IF SECOND = FALSE THEN
        IF CHANGE THEN SECOND = TRUE: GOTO newcalc   'IF WE'VE REJECT ANY
NEW GO CALC AGAIN
      END IF
'next GAS
     NEXT GAS
     DIFFERENCE# = GAS#(1, CYCLE) - GAS#(2, CYCLE)
     IF AUTO.HARDCOPY THEN GOSUB print.cycle
     IF SCALE.CALCULATED = FALSE THEN GOSUB scaling.routine
     GOSUB print.cycle.to.screen
'next CYCLE
    NEXT CYCLE   'GO DO NEXT CYCLE
    KEY(1) OFF
    GOSUB calculate.final.values
    GOSUB calculate.del.factors
    DIFFERENCE# = GAS.MEAN#(1) - GAS.MEAN#(2)
    GOSUB save.raw.data
    IF AUTO.HARDCOPY THEN GOSUB print.final.statistics
    VIEW PRINT
    IF AUTO.HARDCOPY THEN
        GOSUB remove.data
        GOSUB screen.dump
    END IF
    NO.DATA = FALSE
    GOSUB select.valve1
    ClearWindow
    GOSUB header
    GOSUB footer
    RETURN

blank.menu:
    FOR i = 1 TO 17 STEP 2
        LOCATE 5 + i, 10
        PRINT SPC(18);
    NEXT i
    RETURN
```

```
blank.setup:
    FOR i = 1 TO 9
        LOCATE 10 + i, 10: PRINT SPC(69);
    NEXT i
    LOCATE 24, 1: PRINT SPC(79);
    RETURN

calc.per.cycle.and.gas:
     'LOOPCOUNT = LOOPCOUNT + 1
      TOT.INT.1 = 0        'ZERO TOTAL READINGS FOR RATIO 45/44
      TOT.INT.2 = 0        'ZERO TOTAL READINGS FOR RATIO 46/44
      TOTAL.1# = 0#         'ZERO SUMS FOR RATIOS
      TOTAL.2# = 0#
      REJECT.1(GAS, CYCLE) = 0  'ZERO NUMBER OF REJECTED READINGS
      REJECT.2(GAS, CYCLE) = 0
    '
    '  SUM READINGS FOR MEAN CALCULATION FOR THOSE READINGS WHICH
    '  WERE NOT REJECTED.
    '
      FOR READING = 1 TO TOTAL.INTEGRATIONS
        IF DEL.1(GAS, CYCLE, READING) THEN
            REJECT.1(GAS, CYCLE) = REJECT.1(GAS, CYCLE) + 1
        ELSE
            TOT.INT.1 = TOT.INT.1 + 1
            TOTAL.1# = TOTAL.1# + IR.READ.1#(GAS, CYCLE, READING)
        END IF
        IF DEL.2(GAS, CYCLE, READING) THEN
            REJECT.2(GAS, CYCLE) = REJECT.2(GAS, CYCLE) + 1
        ELSE
            TOT.INT.2 = TOT.INT.2 + 1
            TOTAL.2# = TOTAL.2# + IR.READ.2#(GAS, CYCLE, READING)
        END IF
      NEXT READING
      MEAN.1#(GAS, CYCLE) = TOTAL.1# / TOT.INT.1    'CALCULATE MEANS
      MEAN.2#(GAS, CYCLE) = TOTAL.2# / TOT.INT.2
    '
    '  CALCULATE STANTARD DEVIATION USING ONLY THOSE READINGS
    '  WHICH WERE NOT REJECTED.
    '
      SD.1#(GAS, CYCLE) = 0#
      SD.2#(GAS, CYCLE) = 0#
      FOR READING = 1 TO TOTAL.INTEGRATIONS
         IF TOT.INT.1 <> 1 THEN
             IF DEL.1(GAS, CYCLE, READING) = FALSE THEN
                 SD.1#(GAS, CYCLE) = SD.1#(GAS, CYCLE) + ((IR.READ.1#(GAS,
CYCLE, READING) - MEAN.1#(GAS, CYCLE)) ^ 2)
             END IF
         END IF
         IF TOT.INT.2 <> 1 THEN
             IF DEL.2(GAS, CYCLE, READING) = FALSE THEN
```

```
                    SD.2#(GAS, CYCLE) = SD.2#(GAS, CYCLE) + ((IR.READ.2#(GAS,
CYCLE, READING) - MEAN.2#(GAS, CYCLE)) ^ 2)
                END IF
            END IF
        NEXT READING
        IF TOT.INT.1 <> 1 THEN
            SD.1#(GAS, CYCLE) = SD.1#(GAS, CYCLE) / (TOT.INT.1 - 1)
            SD.1#(GAS, CYCLE) = SD.1#(GAS, CYCLE) ^ .5
        END IF
        IF TOT.INT.2 <> 1 THEN
            SD.2#(GAS, CYCLE) = SD.2#(GAS, CYCLE) / (TOT.INT.2 - 1)
            SD.2#(GAS, CYCLE) = SD.2#(GAS, CYCLE) ^ .5
            '
            '  WE NOW MARK THOSE READINGS THAT ARE "DEVIATIONS
ALLOWED"
            '  STANDARD DEVIATIONS FROM OUR MEAN.
            '
            FOR READING = 1 TO TOTAL.INTEGRATIONS
                IF ABS(MEAN.1#(GAS, CYCLE) - IR.READ.1#(GAS, CYCLE, READING)) >
READING.REJECT! * SD.1#(GAS, CYCLE) THEN DEL.1(GAS, CYCLE, READING) =
TRUE: CHANGE = TRUE
                IF ABS(MEAN.2#(GAS, CYCLE) - IR.READ.2#(GAS, CYCLE, READING)) >
READING.REJECT! * SD.2#(GAS, CYCLE) THEN DEL.2(GAS, CYCLE, READING) =
TRUE: CHANGE = TRUE
            NEXT READING
            '
            '   IF WE HAVE ALREADY RECALCULATED ONCE, THEN WE WILL NOT
            '   ANYMORE.  WE DO NOT WANT TO GET LOCKED INTO DELETING
            '   ALL BUT ONE READING
            '
        END IF
        RETURN

calculate.del.factors:
      DEL.TOT.1 = 0
      DEL.TOT.2 = 0
      DEL#(1) = 0
      DEL#(2) = 0
      DEVIATION#(1) = 0
      DEVIATION#(2) = 0
      FOR CYCLE = 1 TO CYCLES
        IF CYCLES.DEL.1(CYCLE) = FALSE THEN
            DEL1#(CYCLE) = ((MEAN.1#(2, CYCLE) - MEAN.1#(1, CYCLE)) / (MEAN.1#(1,
CYCLE))) * 1000
            DEL#(1) = DEL#(1) + DEL1#(CYCLE)
            DEL.TOT.1 = DEL.TOT.1 + 1
        END IF
        IF CYCLES.DEL.2(CYCLE) = FALSE THEN
            DEL2#(CYCLE) = ((MEAN.2#(2, CYCLE) - MEAN.2#(1, CYCLE)) / (MEAN.2#(1,
CYCLE))) * 1000
            DEL#(2) = DEL#(2) + DEL2#(CYCLE)
            DEL.TOT.2 = DEL.TOT.2 + 1
```

```
    END IF
  NEXT CYCLE
  DEL#(1) = DEL#(1) / DEL.TOT.1
  DEL#(2) = DEL#(2) / DEL.TOT.2
  '
  '  now calculate the standard deviation of our Dels
  '
  FOR CYCLE = 1 TO CYCLES
   IF CYCLES.DEL.1(CYCLE) = FALSE THEN
       DEVIATION#(1) = DEVIATION#(1) + ((DEL1#(CYCLE) - DEL#(1)) ^ 2)
   END IF
    IF CYCLES.DEL.2(CYCLE) = FALSE THEN
       DEVIATION#(2) = DEVIATION#(2) + ((DEL2#(CYCLE) - DEL#(2)) ^ 2)
   END IF
  NEXT CYCLE
  DEVIATION#(1) = (DEVIATION#(1) / (DEL.TOT.1 - 1)) ^ .5
  DEVIATION#(2) = (DEVIATION#(2) / (DEL.TOT.2 - 1)) ^ .5
  '
  '   CALCULATE THE STANDARD ERROR OF OUR DEL FACTORS
  '
  DEL.ERROR#(1) = DEVIATION#(1) / SQR(DEL.TOT.1)
  DEL.ERROR#(2) = DEVIATION#(2) / SQR(DEL.TOT.2)
  RETURN

calculate.final.values:
  GAS.MEAN#(1) = GAS.MEAN#(1) / CYCLES   ' calculate mean for gas1 denom
  GAS.MEAN#(2) = GAS.MEAN#(2) / CYCLES   ' calculate mean for gas2 denom
  '
  '   Calculate standard deviation of our major isotopes
  '   and the final means for each gas.
  '
  FINAL.MEAN.1#(1) = 0
  FINAL.MEAN.1#(2) = 0
  FINAL.MEAN.2#(1) = 0
  FINAL.MEAN.2#(2) = 0
  TOT.1 = 0
  TOT.2 = 0
  GAS.DEV#(1) = 0!
  GAS.DEV#(2) = 0!
  FOR CYCLE = 1 TO CYCLES
    FINAL.MEAN.1#(1) = FINAL.MEAN.1#(1) + MEAN.1#(1, CYCLE)
    FINAL.MEAN.1#(2) = FINAL.MEAN.1#(2) + MEAN.1#(2, CYCLE)
    FINAL.MEAN.2#(1) = FINAL.MEAN.2#(1) + MEAN.2#(1, CYCLE)
    FINAL.MEAN.2#(2) = FINAL.MEAN.2#(2) + MEAN.2#(2, CYCLE)
    GAS.DEV#(1) = GAS.DEV#(1) + ((GAS#(1, CYCLE) - GAS.MEAN#(1)) ^ 2)
    GAS.DEV#(2) = GAS.DEV#(2) + ((GAS#(2, CYCLE) - GAS.MEAN#(2)) ^ 2)
  NEXT CYCLE
  FINAL.MEAN.1#(1) = FINAL.MEAN.1#(1) / CYCLES
  FINAL.MEAN.1#(2) = FINAL.MEAN.1#(2) / CYCLES
  FINAL.MEAN.2#(1) = FINAL.MEAN.2#(1) / CYCLES
  FINAL.MEAN.2#(2) = FINAL.MEAN.2#(2) / CYCLES
  GAS.DEV#(1) = GAS.DEV#(1) / (CYCLES - 1)
```

```
    GAS.DEV#(1) = GAS.DEV#(1) ^ .5
    GAS.DEV#(2) = GAS.DEV#(2) / (CYCLES - 1)
    GAS.DEV#(2) = GAS.DEV#(2) ^ .5
    '
    '    Calculate Final Standard Deviation For Each Gas
    '
    FINAL.DEV.1#(1) = 0
    FINAL.DEV.1#(2) = 0
    FINAL.DEV.2#(1) = 0
    FINAL.DEV.2#(2) = 0
    FOR i = 1 TO CYCLES
      FOR G = 1 TO 2
        FINAL.DEV.1#(G) = FINAL.DEV.1#(G) + ((MEAN.1#(G, i) - FINAL.MEAN.1#(G))
^ 2)
        FINAL.DEV.2#(G) = FINAL.DEV.2#(G) + ((MEAN.2#(G, i) - FINAL.MEAN.2#(G))
^ 2)
      NEXT G
    NEXT i
    FOR G = 1 TO 2
      FINAL.DEV.1#(G) = FINAL.DEV.1#(G) / (CYCLES - 1)
      FINAL.DEV.1#(G) = FINAL.DEV.1#(G) ^ .5
      FINAL.DEV.2#(G) = FINAL.DEV.2#(G) / (CYCLES - 1)
      FINAL.DEV.2#(G) = FINAL.DEV.2#(G) ^ .5
    NEXT G
    '
    '   See if we should delete any of our cycles
    '
    TOT.1 = 0
    TOT.2 = 0
    FOR i = 1 TO CYCLES
      FOR G = 1 TO 2
        IF (ABS(MEAN.1#(G, i) - FINAL.MEAN.1#(G))) > (CYCLES.REJECT! *
FINAL.DEV.1#(G)) THEN CYCLES.DEL.1(i) = TRUE ELSE CYCLES.DEL.1(i) = FALSE:
TOT.1 = TOT.1 + 1
        IF (ABS(MEAN.2#(G, i) - FINAL.MEAN.2#(G))) > (CYCLES.REJECT! *
FINAL.DEV.2#(G)) THEN CYCLES.DEL.2(i) = TRUE ELSE CYCLES.DEL.2(i) = FALSE:
TOT.2 = TOT.2 + 1
       NEXT G
    NEXT i
    IF TOT.1 <> CYCLES GOTO 169    ' Have a deletion
    IF TOT.2 = CYCLES GOTO 172     ' no changes have taken place
    '
169 :
    '   Have a deletion to take palce
    '
    FINAL.MEAN.1#(1) = 0
    FINAL.MEAN.1#(2) = 0
    FINAL.MEAN.2#(1) = 0
    FINAL.MEAN.2#(2) = 0
    TOT.1 = 0
    TOT.2 = 0
    FOR CYCLE = 1 TO CYCLES
```

```
        IF CYCLES.DEL.1(CYCLE) = FALSE THEN FINAL.MEAN.1#(1) =
FINAL.MEAN.1#(1) + MEAN.1#(1, CYCLE): TOT.1 = TOT.1 + 1
        IF CYCLES.DEL.1(CYCLE) = FALSE THEN FINAL.MEAN.1#(2) =
FINAL.MEAN.1#(2) + MEAN.1#(2, CYCLE)
        IF CYCLES.DEL.2(CYCLE) = FALSE THEN FINAL.MEAN.2#(1) =
FINAL.MEAN.2#(1) + MEAN.2#(1, CYCLE): TOT.2 = TOT.2 + 1
        IF CYCLES.DEL.2(CYCLE) = FALSE THEN FINAL.MEAN.2#(2) =
FINAL.MEAN.2#(2) + MEAN.2#(2, CYCLE)
    NEXT CYCLE
    FINAL.MEAN.1#(1) = FINAL.MEAN.1#(1) / TOT.1
    FINAL.MEAN.1#(2) = FINAL.MEAN.1#(2) / TOT.1
    FINAL.MEAN.2#(1) = FINAL.MEAN.2#(1) / TOT.2
    FINAL.MEAN.2#(2) = FINAL.MEAN.2#(2) / TOT.2
    FINAL.DEV.1#(1) = 0
    FINAL.DEV.1#(2) = 0
    FINAL.DEV.2#(1) = 0
    FINAL.DEV.2#(2) = 0
    FOR i = 1 TO CYCLES
      FOR G = 1 TO 2
        IF CYCLES.DEL.1(i) = FALSE THEN FINAL.DEV.1#(G) = FINAL.DEV.1#(G) +
((MEAN.1#(G, i) - FINAL.MEAN.1#(G)) ^ 2)
        IF CYCLES.DEL.2(i) = FALSE THEN FINAL.DEV.2#(G) = FINAL.DEV.2#(G) +
((MEAN.2#(G, i) - FINAL.MEAN.2#(G)) ^ 2)
      NEXT G
    NEXT i
    FOR G = 1 TO 2
      FINAL.DEV.1#(G) = FINAL.DEV.1#(G) / (TOT.1 - 1)
      FINAL.DEV.1#(G) = (FINAL.DEV.1#(G)) ^ .5
172 :
      FINAL.DEV.2#(G) = FINAL.DEV.2#(G) / (TOT.2 - 1)
      FINAL.DEV.2#(G) = (FINAL.DEV.2#(G)) ^ .5
    NEXT G
    A.1$ = "Cycles Rejected:"
    A.2$ = "Cycles Rejected:"
    FOR CYCLE = 1 TO CYCLES
       IF CYCLES.DEL.1(CYCLE) THEN A.1$ = A.1$ + MID$(STR$(CYCLE), 2,
LEN(STR$(CYCLE)) - 1) + ","
       IF CYCLES.DEL.2(CYCLE) THEN A.2$ = A.2$ + MID$(STR$(CYCLE), 2,
LEN(STR$(CYCLE)) - 1) + ","
    NEXT CYCLE
    RETURN

check.responses:
    response.valid = FALSE
    IF upper.limit!(line.number) = 0 THEN
        IF setup.response$(line.number) = "N" OR setup.response$(line.number) = "Y" THEN
            response.valid = TRUE
        END IF
    ELSEIF VAL(setup.response$(line.number)) >= lower.limit!(line.number) THEN
        IF VAL(setup.response$(line.number)) <= upper.limit!(line.number) THEN
            response.valid = TRUE
        END IF
```

```
	END IF
	RETURN


data.hardcopy:
	GOSUB print.header
	FOR CYCLE = 1 TO CYCLES
		DIFFERENCE# = GAS#(1, CYCLE) - GAS#(2, CYCLE)
		GOSUB print.cycle
	NEXT CYCLE
	GOSUB print.final.statistics
	BEEP
	RETURN


edit.setup:
	CALL SetCurrentWindow(2)
	ClearWindow
	GOSUB blank.menu
	GOSUB print.menu
	line.number = 1
	COLOR 7: LOCATE 24, 1
	PRINT CHR$(24); CHR$(25); : COLOR 10: PRINT "<"; : COLOR 7: PRINT "CR";
	COLOR 10: PRINT "> to select     <"; : COLOR 7: PRINT "F1"; : COLOR 10
	PRINT "> to save   <"; : COLOR 7: PRINT "F2"; : COLOR 10: PRINT "> to continue   <";
	COLOR 7: PRINT "F3"; : COLOR 10: PRINT "> to abandon changes"
	COLOR 7
	GOSUB get.data
	CALL SetCurrentWindow(1)
	GOSUB footer
	RETURN


error.trap:
	BEEP
	IF ERR = 53 THEN
		LOCATE 10, 45: PRINT "Data file does not exist!";
		a$ = INPUT$(1)
		RESUME bailout
	ELSE
		LOCATE 10, 45: PRINT "Error "; ERR; " has occured";
		a$ = INPUT$(1)
		RESUME bailout
	END IF
	RETURN


footer:
	LINE (0, 440)-(639, 440)
	LOCATE 29, 25: PRINT "Current setup: "; setup.name$;
	RETURN
```

```
form.feed:
      PRINT #3, CHR$(12);
      RETURN

get.cfg.list:
      SHELL "dir *.stu|sort/+25>files.cfg"
      'SHELL "type files.cfg"
      OPEN "files.cfg" FOR INPUT AS #4
      filecount = 0
      WHILE NOT EOF(4)
            INPUT #4, dirlist$(filecount)
            IF INSTR(dirlist$(filecount), "STU") = 10 THEN
                  dirlist$(filecount) = LEFT$(dirlist$(filecount), 8)
                  filecount = filecount + 1
            ELSE dirlist$(filecount) = ""
            END IF
      WEND
      CLOSE #4
      SHELL "del files.cfg"
      RETURN

get.data:
    a$ = ""              'while not F1, F2 or F3
    WHILE a$ <> CHR$(27)
            a$ = INKEY$
      IF LEN(a$) = 1 THEN
            IF a$ = CHR$(13) THEN
                  line.number = line.number + 1
                  IF line.number = 10 THEN line.number = 1
                  GOSUB print.responses
            ELSEIF (ASC(a$) > 47 AND ASC(a$) < 58) OR UCASE$(a$) = "N" OR
UCASE$(a$) = "Y" OR a$ = "." THEN
                  IF a$ = "n" THEN a$ = "N"
                  IF a$ = "y" THEN a$ = "Y"
                  setup.response$(line.number) = a$
                  GOSUB print.responses
                  IF a$ = "N" OR a$ = "Y" THEN a$ = CHR$(13)
                  WHILE a$ <> CHR$(13)
                        a$ = INPUT$(1)
                        IF a$ = CHR$(8) AND LEN(setup.response$(line.number)) >= 1 THEN
                              strlength = LEN(setup.response$(line.number))
                              setup.response$(line.number) = LEFT$(setup.response$(line.number),
strlength - 1)
                              GOSUB print.responses
                        ELSEIF (ASC(a$) > 47 AND ASC(a$) < 58) OR a$ = "." THEN
                              setup.response$(line.number) = setup.response$(line.number) + a$
                              GOSUB print.responses
                        ELSEIF a$ <> CHR$(13) THEN
                              BEEP
                        END IF
                  WEND
```

```
            GOSUB check.responses
            IF response.valid = TRUE THEN
                IF line.number = 2 OR line.number = 4 THEN
                    GOSUB show.chauvenet
                END IF
                line.number = line.number + 1
                IF line.number = 10 THEN line.number = 1
            ELSE
                setup.response$(line.number) = "range error"
                BEEP
            END IF
            GOSUB print.responses
            setup.edited = TRUE
        ELSE
            BEEP
        END IF
    ELSEIF RIGHT$(a$, 1) = CHR$(72) THEN
        line.number = line.number - 1
        IF line.number = 0 THEN line.number = 9
        GOSUB print.responses
    ELSEIF RIGHT$(a$, 1) = CHR$(80) THEN
        line.number = line.number + 1
        IF line.number = 10 THEN line.number = 1
        GOSUB print.responses
    ELSEIF RIGHT$(a$, 1) = CHR$(59) THEN        'F1
        GOSUB save.setup
        setup.edited = FALSE
        a$ = CHR$(27)
    ELSEIF RIGHT$(a$, 1) = CHR$(60) THEN        'F2
        a$ = CHR$(27)
    ELSEIF RIGHT$(a$, 1) = CHR$(61) THEN        'F3
        FOR i = 1 TO 9
            setup.response$(i) = save.response$(i)
        NEXT i
        setup.edited = FALSE
        a$ = CHR$(27)
    END IF
   WEND
   GOSUB blank.setup
   GOSUB pass.variables
   ClearWindow
   RETURN

HALT:
    t! = TIMER
    t1! = TIMER
    WHILE (t1! - t!) < delay
        t1! = TIMER
    WEND
    RETURN

HALT2:
```

```
    t! = TIMER
    t1! = TIMER
    WHILE (t1! - t!) < delay2 / 1000
        t1! = TIMER
    WEND
    RETURN


header:
    LOCATE 1, 20: COLOR 3: PRINT "MASS";
    COLOR 7
    PRINT "...Carbon Isotope Mass Spec Controller"
    LOCATE 2, 27: PRINT "Copyright 1991, Government of Canada"
    LINE (0, 40)-(639, 40)
    RETURN



initialize:
    CLS
    SCREEN 9
    valve$ = "Valve 1 Selected"
    TRUE = 1
    FALSE = 0
    STARTUP = TRUE          'haven't talked to mass spectrometer
    setup.edited = FALSE
    setup.name$ = "DEFAULT"

    OPEN "printer.cfg" FOR INPUT AS #5
    LINE INPUT #5, printer$
    LINE INPUT #5, ioport$
    CLOSE #5

    WIDTH ioport$, 120
    OPEN ioport$ FOR OUTPUT AS #3
    SELECT CASE printer$
        CASE "HPLaser"
            PRINT #3, CHR$(27); "(s16.66H"; 'Line Prntr, 16.66 cpi
        CASE "PR-1012"
            PRINT #3, CHR$(27); "w2";      '15 cpi
        CASE "Epson LQ"
            PRINT #3, CHR$(27); "g";       '15 cpi
    END SELECT

    OPEN "default.stu" FOR INPUT AS #5
    line.number = 1
    WHILE NOT EOF(5)
        LINE INPUT #5, line$
        setup.line$(line.number) = LEFT$(line$, 48)
        setup.response$(line.number) = MID$(line$, 50, LEN(line$) - 49)
        save.response$(line.number) = setup.response$(line.number)
        line.number = line.number + 1
```

```
    WEND
    CLOSE #5
    GOSUB pass.variables
    blank$ = STRING$(80, " ")
    NO.DATA = TRUE
    RETURN

init.IEEE.488:
        'LOCATE 1, 1: PRINT "init.IEEE.488": a$ = INPUT$(1): LOCATE 1, 1: PRINT
SPC(20);
    ' ************************************************************
    '
    '       INITIALIZATION OF THE IEEE 488 CARD
    '
    '
    KEY(1) STOP
    BAS% = &H310
    CARD% = 21
    NUM2% = 15
    OUT (BAS% + 9), &HF2: GOSUB pause
    OUT (BAS% + 5), 2
    OUT (BAS% + 1), 3
    OUT (BAS% + 2), 0
    OUT (BAS% + 3), 0
    OUT (BAS% + 4), 1
    OUT (BAS% + 5), &H90
    OUT (BAS% + 5), &HA0
    OUT (BAS% + 6), 1
    OUT (BAS% + 6), &HE0
    OUT (BAS% + 7), &HD
    OUT (BAS% + 8), &H80
    OUT (BAS% + 8), 0
    OUT (BAS% + 9), &HFF: GOSUB pause
    OUT (BAS% + 5), 0
    STAT% = INP(BAS% + 1)
    OUT (BAS% + 9), &HFD: GOSUB pause
    OUT (BAS% + 4), &H80
    OUT (BAS% + 5), 0
    WAIT (BAS% + 1), 2
    OUT (BAS% + 0), &H5F
    WAIT (BAS% + 1), 2
    OUT (BAS% + 0), &H3F
    OUT (BAS% + 5), 0
    DEV% = 4
    OUT (BAS% + 4), &H40
    OUT (BAS% + 6), DEV%
    WAIT (BAS% + 1), 2
    OUT (BAS% + 0), (DEV% + &H20)
    OUT (BAS% + 4), &H80
    OUT (BAS% + 6), CARD%
    OUT (BAS% + 5), 0
    WAIT (BAS% + 1), 2
```

```
    OUT (BAS% + 0), &H4
    KEY(1) ON
    RETURN


IR.output.routine:
        'LOCATE 2, 1: PRINT "IR.output.routine": a$ = INPUT$(1): LOCATE 2, 1: PRINT
SPC(20);
    ' *******************************************************************
    '
    '
    '       OUTPUT ROUTINE FOR IR
    '
    KEY(1) STOP
    DEV% = 4
    STAT% = INP(BAS% + 1)
    OUT (BAS% + 9), &HFD: GOSUB pause
    OUT (BAS% + 4), &H80
    OUT (BAS% + 5), 0
    WAIT (BAS% + 1), 2
    OUT (BAS% + 0), &H5F
    WAIT (BAS% + 1), 2
    OUT (BAS% + 0), &H3F
    OUT (BAS% + 4), &H40
    OUT (BAS% + 6), DEV%
    WAIT (BAS% + 1), 2
    OUT (BAS% + 0), (DEV% + &H20)
    OUT (BAS% + 4), &H80
    OUT (BAS% + 6), CARD%
    OUT (BAS% + 5), 0
    OUT (BAS% + 9), &HF6: GOSUB pause
    IR.OUT$ = IR.OUT$ + CHR$(13)
    N = LEN(IR.OUT$)
    FOR i = 1 TO N
      a = ASC(MID$(IR.OUT$, i, 1))
      WAIT (BAS% + 1), 2
      OUT (BAS% + 0), a
    NEXT i
    OUT (BAS% + 9), &HFD: GOSUB pause
    KEY(1) ON
    RETURN

load.setup:
    CALL SetCurrentWindow(2)
    delete.flag = FALSE
    GOSUB show.directory
    menu = 2: item = 1: maxitem = filecount: GOSUB selection
    setup.name$ = UCASE$(RTRIM$(dirlist$(item - 1)))
    IF delete.flag THEN
        delete.flag = FALSE
        IF setup.name$ = "DEFAULT" THEN
            LOCATE 9 + filecount, 50
```

```
                BEEP
                PRINT "Can't delete DEFAULT!!";
                a$ = INPUT$(1)
            ELSE
                LOCATE 9 + filecount, 50
                PRINT "DELETING "; setup.name$; "....";
                SHELL "del " + RTRIM$(setup.name$) + ".stu"
                GOSUB get.cfg.list
                GOSUB footer
            END IF
            ClearWindow
        ELSE
            OPEN setup.name$ + ".stu" FOR INPUT AS #5
            line.number = 1
            WHILE NOT EOF(5)
                LINE INPUT #5, line$
                setup.line$(line.number) = LEFT$(line$, 48)
                setup.response$(line.number) = MID$(line$, 50, LEN(line$) - 49)
                save.response$(line.number) = setup.response$(line.number)
                line.number = line.number + 1
            WEND
            CLOSE #5
            setup.edited = FALSE
            GOSUB edit.setup
        END IF
        menu = 1: maxitems = 9: item = 1
        RETURN

menu1:
        LOCATE 6, 10
        FOR i = 1 TO 11
            IF i = item THEN
                PRINT CHR$(26);
                COLOR 12
            ELSE
                PRINT CHR$(32);
                COLOR 7
            END IF
            PRINT item$(i): PRINT
            COLOR 7
            LOCATE , 10
        NEXT i
        RETURN

menu2:
        LOCATE 7, 50
        FOR i = 1 TO maxitems
            IF i = item THEN
                PRINT CHR$(26);
                COLOR 12
            ELSE
                PRINT CHR$(32);
```

```
                COLOR 7
            END IF
            PRINT dirlist$(i - 1)
            COLOR 7
            LOCATE , 50
        NEXT i
        RETURN

pass.variables:
        RANGE = VAL(setup.response$(1))
        TOTAL.INTEGRATIONS = VAL(setup.response$(2))
        READING.REJECT! = VAL(setup.response$(3))
        CYCLES = VAL(setup.response$(4))
        CYCLES.REJECT! = VAL(setup.response$(5))
        delay = (VAL(setup.response$(6)))
        IF setup.response$(7) = "N" THEN AUTOSCALE = FALSE ELSE AUTOSCALE = TRUE
        FIXED.SCALE! = (VAL(setup.response$(8))) / 1000000
        IF setup.response$(9) = "N" THEN AUTO.HARDCOPY = FALSE ELSE
AUTO.HARDCOPY = TRUE
        RETURN

pause:
        FOR NUM1 = 1 TO 10 * NUM2
        NEXT NUM1
        RETURN

plot.data:
        IF SCALE.CALCULATED = FALSE THEN RETURN 'not ready yet
        LAST.Y = LAST.Y + 1
        X1 = 62 + INT((NUMERATOR.1# - START.VOLTS.1#) / VOLTS.PER.DOT.1#)
        IF X1 < 0 THEN X1 = 0
        IF X1 > 309 THEN X1 = 309
        X2 = 382 + INT((NUMERATOR.2# - START.VOLTS.2#) / VOLTS.PER.DOT.2#)
        IF X2 > 639 THEN X2 = 639
        IF X2 < 320 THEN X2 = 320
        PSET (X1, INT(LAST.Y / scale.factor))
        PSET (X2, INT(LAST.Y / scale.factor))
        IF LINE.PLOT THEN
            LINE (LAST.X1, INT((LAST.Y - 1) / scale.factor))-(X1, INT((LAST.Y - 1) /
scale.factor))
            LINE (LAST.X2, INT((LAST.Y - 1) / scale.factor))-(X2, INT((LAST.Y - 1) /
scale.factor))
        END IF
        IF READING = TOTAL.INTEGRATIONS THEN LINE.PLOT = TRUE: LAST.X1 = X1:
LAST.X2 = X2 ELSE LINE.PLOT = FALSE
        RETURN


plot.hardcopy:
        GOSUB blank.menu
        SetCurrentWindow (1)
        ClearWindow
```

```
    LOCATE 1, 1: PRINT D$; SPC(20); title$, ; SPC(20); t$
    LINE.PLOT = FALSE        'SET PLOTTING FLAG FALSE
    LAST.Y = 25
    FOR CYCLE = 1 TO CYCLES
     FOR GAS = 1 TO 2
      FOR READING = 1 TO TOTAL.INTEGRATIONS
        NUMERATOR.1# = IR.READ.1#(GAS, CYCLE, READING)
        NUMERATOR.2# = IR.READ.2#(GAS, CYCLE, READING)
        GOSUB plot.data
      NEXT READING
     NEXT GAS
    NEXT CYCLE   'GO DO NEXT CYCLE

    GOSUB screen.dump
    ClearWindow
    GOSUB header
    GOSUB footer
    RETURN

prepare.graphics.screen:
    LINE (1, 240)-(639, 463), 3, B
    LINE (1, 272)-(639, 272), 3
    LINE (1, 274)-(639, 274), 3
    COLOR 2: LOCATE 1, 1: PRINT D$; SPC(20);
    COLOR 3: PRINT title$, ; SPC(20);
    COLOR 2: PRINT t$: COLOR 7
    'LOCATE 3, 5: PRINT "CYCLE: 1";
    'LOCATE 4, 7: PRINT "GAS: 1";
    'LOCATE 5, 3: PRINT "READING: 1";
    COLOR 2: LOCATE 16, 10: PRINT "Mass 45/44";
    LOCATE , 32: PRINT "Mass 46/44";
    LOCATE , 54: PRINT "Mass 44": COLOR 7
    LOCATE , 2: PRINT "CYC    Gas1     Gas2     Gas1     Gas2     Gas1     Gas2
Diff"
    LINE (1, 240)-(639, 463), 3, B
    LOCATE 30, 10: PRINT "F1 to abort gas analysis";
    LOCATE 19, 2
    RETURN


printer.config:
    LINE (375, 342)-(528, 375), , B
    LOCATE 22, 50
    PRINT "<ESC> to accept"
    LOCATE , 50: COLOR 12
    PRINT printer$: COLOR 7
    LOCATE , 50
    PRINT "<CR> to change"
    a$ = INKEY$
    WHILE a$ <> CHR$(27)
        IF a$ = CHR$(13) THEN
            IF printer$ = "HPLaser" THEN
```

```
                printer$ = "PR-1012"
            ELSEIF printer$ = "PR-1012" THEN
                printer$ = "Epson LQ"
            ELSE
                printer$ = "HPLaser"
            END IF
            LOCATE 23, 50: COLOR 12: PRINT printer$; " "; : COLOR 7
        END IF
        a$ = INKEY$
    WEND
    LOCATE 23, 50: COLOR 12
    PRINT ioport$; "     "; : COLOR 7
    a$ = INKEY$
    WHILE a$ <> CHR$(27)
        IF a$ = CHR$(13) THEN
            IF ioport$ = "LPT1:" THEN
                ioport$ = "LPT2:"
            ELSE
                ioport$ = "LPT1:"
            END IF
            LOCATE 23, 50: COLOR 12: PRINT ioport$: COLOR 7
        END IF
        a$ = INKEY$
    WEND
    FOR i = 1 TO 5: LOCATE 19 + i, 40: PRINT SPC(29); : NEXT i

    OPEN "printer.cfg" FOR OUTPUT AS #5
    PRINT #5, printer$
    PRINT #5, ioport$
    CLOSE #5

    CLOSE #3
    WIDTH ioport$, 120
    OPEN ioport$ FOR OUTPUT AS #3
    SELECT CASE printer$
        CASE "HPLaser"
            PRINT #3, CHR$(27); "(s16.66H"; 'Line Prntr, 16.66 cpi
        CASE "PR-1012"
            PRINT #3, CHR$(27); "w2";     '15 cpi
        CASE "Epson LQ"
            PRINT #3, CHR$(27); "g";      '15 cpi
    END SELECT

    RETURN

print.cycle:
    PRINT #3, TAB(1); USING "##"; CYCLE;
    PRINT #3, TAB(6); USING "#.#####"; MEAN.1#(1, CYCLE);
    PRINT #3, TAB(15); USING "#.#####"; SD.1#(1, CYCLE);
    PRINT #3, TAB(23); USING "#"; REJECT.1(1, CYCLE);
    PRINT #3, TAB(25); USING "#.#####"; MEAN.1#(2, CYCLE);
    PRINT #3, TAB(35); USING "#.#####"; SD.1#(2, CYCLE);
```

```
      PRINT #3, TAB(43); USING "#"; REJECT.1(2, CYCLE);
      PRINT #3, TAB(45); USING "#.#####"; MEAN.2#(1, CYCLE);
      PRINT #3, TAB(54); USING "#.#####"; SD.2#(1, CYCLE);
      PRINT #3, TAB(62); USING "#"; REJECT.2(1, CYCLE);
      PRINT #3, TAB(64); USING "#.#####"; MEAN.2#(2, CYCLE);
      PRINT #3, TAB(73); USING "#.#####"; SD.2#(2, CYCLE);
      PRINT #3, TAB(81); USING "#"; REJECT.2(2, CYCLE);
      PRINT #3, TAB(85); USING "##.##"; GAS#(1, CYCLE);
      PRINT #3, TAB(92); USING "##.##"; GAS#(2, CYCLE);
      PRINT #3, TAB(99); USING "###.###"; DIFFERENCE#
      RETURN


print.cycle.to.screen:
      cursor.vert.position = CSRLIN
'     VIEW PRINT
'     LOCATE 3, 12: PRINT CYCLE;
'     VIEW PRINT 19 TO 28
      IF cursor.vert.position < 28 THEN
         LOCATE cursor.vert.position, 2      '+1
      ELSE
         LOCATE 28, 2
      END IF
      PRINT USING "##"; CYCLE;
      LOCATE , 8: PRINT USING "#.#####"; MEAN.1#(1, CYCLE);
      LOCATE , 19: PRINT USING "#.#####"; MEAN.1#(2, CYCLE);
      LOCATE , 30: PRINT USING "#.#####"; MEAN.2#(1, CYCLE);
      LOCATE , 41: PRINT USING "#.#####"; MEAN.2#(2, CYCLE);
      LOCATE , 52: PRINT USING "##.###"; GAS#(1, CYCLE);
      LOCATE , 63: PRINT USING "##.###"; GAS#(2, CYCLE);
      LOCATE , 72: PRINT USING "###.###"; DIFFERENCE#;
      IF CYCLE <> CYCLES THEN PRINT
      LINE (1, 240)-(639, 463), 3, B
      RETURN

print.final.statistics:
      PRINT #3, " "
      PRINT #3, TAB(1); "FIN"; TAB(6); USING "#.#####"; FINAL.MEAN.1#(1);
      PRINT #3, TAB(15); USING "#.#####"; FINAL.DEV.1#(1);
      PRINT #3, TAB(25); USING "#.#####"; FINAL.MEAN.1#(2);
      PRINT #3, TAB(35); USING "#.#####"; FINAL.DEV.1#(2);
      PRINT #3, TAB(45); USING "#.#####"; FINAL.MEAN.2#(1);
      PRINT #3, TAB(54); USING "#.#####"; FINAL.DEV.2#(1);
      PRINT #3, TAB(64); USING "#.#####"; FINAL.MEAN.2#(2);
      PRINT #3, TAB(73); USING "#.#####"; FINAL.DEV.2#(2);
      PRINT #3, TAB(85); USING "##.##"; GAS.MEAN#(1);
      PRINT #3, TAB(92); USING "##.##"; GAS.MEAN#(2);
      PRINT #3, TAB(99); USING "###.###"; DIFFERENCE#
      PRINT #3, " "
      PRINT #3, TAB(17); CHR$(235); "(%.) ="; TAB(25); USING "###.###"; DEL#(1);
      PRINT #3, TAB(56); CHR$(235); "(%.) ="; TAB(64); USING "###.###"; DEL#(2)
```

```
    PRINT #3, " "
    PRINT #3, TAB(17); "s.d.  ="; TAB(25); USING "#.#####"; DEVIATION#(1);
    PRINT #3, TAB(56); "s.d.  ="; TAB(64); USING "#.#####"; DEVIATION#(2)
    PRINT #3, " "
    PRINT #3, TAB(17); "s.e.  ="; TAB(25); USING "#.#####"; DEL.ERROR#(1);
    PRINT #3, TAB(56); "s.e.  ="; TAB(64); USING "#.#####"; DEL.ERROR#(2)
    PRINT #3, " "
    PRINT #3, TAB(12); A.1$; TAB(51); A.2$
    PRINT #3, " "
    a$ = STRING$(96, "_")
    PRINT #3, a$
    PRINT #3, " "
   RETURN


print.header:
    PRINT #3, " "            'PRINT HEADING OF REPORT ON PRINTER
    PRINT #3, " "
    PRINT #3, TAB(5); "DATE :"; TAB(12); USING "\       \"; D$;
    Z = LEN(title$) / 2
    Z = 48 - Z
    PRINT #3, TAB(Z); title$;
    PRINT #3, TAB(76); "TIME :"; TAB(83); USING "\     \"; t$
    PRINT #3, " "
    PRINT #3, " "
    PRINT #3, TAB(20); "Mass 45/44"; TAB(59); "Mass 46/44"; TAB(87); "Mass 44"
    PRINT #3, TAB(20); "----------"; TAB(59); "----------"; TAB(87); "-------"
    PRINT #3, " "
    PRINT #3, " "
    PRINT #3, TAB(1); "CYC"; TAB(7); "Gas 1"; TAB(16); "s.d."; TAB(23); "R"; TAB(26);
"Gas 2"; TAB(36); "s.d."; TAB(43); "R"; TAB(46); "Gas 1"; TAB(55); "s.d."; TAB(62); "R";
TAB(65); "Gas 2"; TAB(74); "s.d."; TAB(81); "R"; TAB(85); "Gas 1"; TAB _
(92); "Gas 2";
    PRINT #3, TAB(1); "---"; TAB(7); "-----"; TAB(16); "----"; TAB(23); "-"; TAB(26); "-----";
TAB(36); "----"; TAB(43); "-"; TAB(46); "-----"; TAB(55); "----"; TAB(62); "-"; TAB(65); "-----
"; TAB(74); "----"; TAB(81); "-"; TAB(85); "-----"; TAB _
(92); "-----";
    RETURN


print.menu:
    FOR line.num = 1 TO 9
        LOCATE 10 + line.num, 10
        PRINT setup.line$(line.num); " ";
        IF line.num = 1 THEN COLOR 14 ELSE COLOR 7
        PRINT setup.response$(line.num); SPC(5);
        COLOR 7
    NEXT line.num
    line.number = 2: GOSUB show.chauvenet
    line.number = 4: GOSUB show.chauvenet
    RETURN
```

```
print.responses:
     FOR line.num = 1 TO 9
          LOCATE 10 + line.num, 59
          IF line.num = line.number THEN COLOR 14 ELSE COLOR 7
          PRINT setup.response$(line.num); SPC(10);
          COLOR 7
     NEXT line.num
     RETURN

recalc.datafile:
     SCALE.CALCULATED = FALSE  'WE HAVE NOT CALCULATED THE SCALE FOR
GRAPH
     GOSUB blank.menu
     SetCurrentWindow (1)
     ClearWindow
     GOSUB prepare.graphics.screen
     LOCATE 30, 10: PRINT SPC(30); 'get rid of "F1 to abort" message
     LOCATE 19, 2
     IF AUTO.HARDCOPY THEN GOSUB print.header
     LINE.PLOT = FALSE        'SET PLOTTING FLAG FALSE
     '
     '  FIRST WE OBTAIN THE READINGS OF OUR TWO NUMERATORS AND
     '  THEN TAKE A RATIO PAIR OF READING, AND THEN DISPLAY THE
     '  DENOMINATOR VALUES FOR OUR TWO GASES.
     '
     GAS.MEAN#(1) = 0!
     GAS.MEAN#(2) = 0!

'start CYCLE
     FOR CYCLE = 1 TO CYCLES
      CYCLES.DEL.1(CYCLE) = FALSE    'SET CYCLE DELETION ARRAYS FALSE
      CYCLES.DEL.2(CYCLE) = FALSE
'start GAS
      FOR GAS = 1 TO 2
       FOR READING = 1 TO TOTAL.INTEGRATIONS
          NUMERATOR.1# = IR.READ.1#(GAS, CYCLE, READING)
          NUMERATOR.2# = IR.READ.2#(GAS, CYCLE, READING)
          GOSUB plot.data
          DEL.1(GAS, CYCLE, READING) = FALSE 'SET DELETED READING FALSE
          DEL.2(GAS, CYCLE, READING) = FALSE
      '   cursor.vert.position = CSRLIN
      '   VIEW PRINT
      '   LOCATE 4, 12: PRINT GAS;
      '   LOCATE 5, 12: PRINT READING;
      '   VIEW PRINT 19 TO 29
      '   LOCATE cursor.vert.position, 2
       NEXT READING
       GAS.MEAN#(GAS) = GAS.MEAN#(GAS) + GAS#(GAS, CYCLE)' sum for mean
       CHANGE = FALSE        'SET READING DELEATED FLAG FALSE
       SECOND = FALSE        'SET SECOND STATISTICS CALCULATED FALSE
newcalc2:
```

```
    GOSUB calc.per.cycle.and.gas
    IF SECOND = FALSE THEN
        IF CHANGE THEN SECOND = TRUE: GOTO newcalc2   'IF WE'VE REJECT
ANY NEW GO CALC AGAIN
     END IF
'next GAS
    NEXT GAS
    DIFFERENCE# = GAS#(1, CYCLE) - GAS#(2, CYCLE)
    IF AUTO.HARDCOPY THEN GOSUB print.cycle
    IF SCALE.CALCULATED = FALSE THEN GOSUB scaling.routine
    GOSUB print.cycle.to.screen

'next CYCLE
   NEXT CYCLE   'GO DO NEXT CYCLE
   GOSUB calculate.final.values
   GOSUB calculate.del.factors
   DIFFERENCE# = GAS.MEAN#(1) - GAS.MEAN#(2)
   IF AUTO.HARDCOPY THEN GOSUB print.final.statistics
   VIEW PRINT
   IF AUTO.HARDCOPY THEN
      GOSUB remove.data
      GOSUB screen.dump
   END IF
   LOCATE 30, 10: BEEP: PRINT ".....PRESS ANY KEY TO CONTINUE";
   a$ = INKEY$
   WHILE a$ = ""
      LOCATE 30, 10
      FOR i = 1 TO 5
         COLOR INT(6 * RND(1))
         PRINT ".";
      NEXT i
      a$ = INKEY$
   WEND
   COLOR 7
   NO.DATA = FALSE
   ClearWindow
   GOSUB header
   GOSUB footer
   RETURN

remove.data:
   LINE (1, 240)-(639, 463), 0, B
   LINE (1, 272)-(639, 272), 0
   LINE (1, 274)-(639, 274), 0
'   LOCATE 3, 5: PRINT SPC(10);
'   LOCATE 4, 3: PRINT SPC(12);
'   LOCATE 5, 3: PRINT SPC(12);
   FOR i = 1 TO 15
      LOCATE 15 + i, 1: PRINT SPC(79);
   NEXT i
   RETURN
```

```
retrieve.datafile:
	SetCurrentWindow (2)
	LINE (340, 156)-(600, 191), , B
	LOCATE 14, 45
	INPUT "Raw Data File Name : ", title$
	IF title$ = "" THEN
		ClearWindow
		LOCATE 10, 45: PRINT SPC(34);
		BEEP
		SetCurrentWindow (1)
		GOSUB footer
		RETURN
	END IF
	IF INSTR(title$, ".") <> 0 THEN title$ = LEFT$(title$, INSTR(title$, ".") - 1)
	exist = FALSE
	ON ERROR GOTO error.trap
	OPEN title$ + ".raw" FOR INPUT AS #6
	ON ERROR GOTO 0
	old.name$ = setup.name$
	LINE INPUT #6, setup.name$
	setup.name$ = UCASE$(setup.name$)
	FOR i = 1 TO filecount
		IF RTRIM$(dirlist$(i - 1)) = LEFT$(setup.name$, LEN(setup.name$) - 4) THEN exist
= TRUE
	NEXT i
	IF exist = FALSE THEN
		BEEP
		LOCATE 10, 45: PRINT "Setup file doesn't exist!  "
		a$ = INPUT$(1)
		LOCATE 10, 45: PRINT SPC(34);
		ClearWindow
		CLOSE #6
		SetCurrentWindow (1)
		setup.name$ = old.name$
		GOSUB footer
		RETURN
	ELSE
		setup.name$ = LEFT$(setup.name$, INSTR(setup.name$, ".") - 1)
		OPEN setup.name$ + ".stu" FOR INPUT AS #7
		line.number = 1
		WHILE NOT EOF(7)
			LINE INPUT #7, line$
			setup.line$(line.number) = LEFT$(line$, 48)
			setup.response$(line.number) = MID$(line$, 50, LEN(line$) - 49)
			line.number = line.number + 1
		WEND
		CLOSE #7
		GOSUB pass.variables    'load origional "girdx" variables
	END IF
	LINE INPUT #6, line$
	D$ = LEFT$(line$, 10)      'date
	line$ = LTRIM$(RIGHT$(line$, LEN(line$) - 10))
```

```
	t$ = LEFT$(line$, 8)	'time
	FOR i = 1 TO 4: LINE INPUT #6, line$: NEXT i
	FOR i = 1 TO CYCLES
		FOR j = 1 TO TOTAL.INTEGRATIONS
			INPUT #6, IR.READ.1#(1, i, j)
		NEXT j
		'LINE INPUT #6, line$
	NEXT i
	FOR i = 1 TO 4: LINE INPUT #6, line$: NEXT i
	FOR i = 1 TO CYCLES
		FOR j = 1 TO TOTAL.INTEGRATIONS
			INPUT #6, IR.READ.2#(1, i, j)
		NEXT j
		INPUT #6, GAS#(1, i)
		'LINE INPUT #6, line$
	NEXT i
	FOR i = 1 TO 4: LINE INPUT #6, line$: NEXT i
	FOR i = 1 TO CYCLES
		FOR j = 1 TO TOTAL.INTEGRATIONS
			INPUT #6, IR.READ.1#(2, i, j)
		NEXT j
		'LINE INPUT #6, line$
	NEXT i
	FOR i = 1 TO 4: LINE INPUT #6, line$: NEXT i
	FOR i = 1 TO CYCLES
		FOR j = 1 TO TOTAL.INTEGRATIONS
			INPUT #6, IR.READ.2#(2, i, j)
		NEXT j
		INPUT #6, GAS#(2, i)
	NEXT i
bailout:	CLOSE #6
	ClearWindow
	LOCATE 10, 45: PRINT SPC(34);
	NO.DATA = TRUE 'no calculations made yet!
	SetCurrentWindow (1)
	GOSUB footer
	RETURN

save.raw.data:
	OPEN title$ + ".raw" FOR OUTPUT AS #4
	PRINT #4, setup.name$; ".STU"
	PRINT #4, DATE$; "   "; TIME$; "      Filename: "; title$; ".raw"
	PRINT #4,
	PRINT #4, "45/44 RATIO....GAS #1"
	PRINT #4,
	FOR READING = 1 TO TOTAL.INTEGRATIONS
		PRINT #4, "readng "; RIGHT$(" " + STR$(READING), 2); "   ";
	NEXT READING
	PRINT #4,
	FOR CYCLE = 1 TO CYCLES
		FOR READING = 1 TO TOTAL.INTEGRATIONS
			PRINT #4, USING "#.######"; IR.READ.1#(1, CYCLE, READING);
```

```
            PRINT #4, "   ";
        NEXT READING
        PRINT #4,
NEXT CYCLE
PRINT #4,
PRINT #4, "46/44 RATIO....GAS #1"
PRINT #4,
FOR READING = 1 TO TOTAL.INTEGRATIONS
        PRINT #4, "readng "; RIGHT$(" " + STR$(READING), 2); "   ";
NEXT READING
PRINT #4, "Mass 44"
FOR CYCLE = 1 TO CYCLES
        FOR READING = 1 TO TOTAL.INTEGRATIONS
            PRINT #4, USING "#.######"; IR.READ.2#(1, CYCLE, READING);
            PRINT #4, "   ";
        NEXT READING
        PRINT #4, USING "##.###"; GAS#(1, CYCLE)
NEXT CYCLE
PRINT #4,
PRINT #4, "45/44 RATIO....GAS #2"
PRINT #4,
FOR READING = 1 TO TOTAL.INTEGRATIONS
        PRINT #4, "readng "; RIGHT$(" " + STR$(READING), 2); "   ";
NEXT READING
PRINT #4,
FOR CYCLE = 1 TO CYCLES
        FOR READING = 1 TO TOTAL.INTEGRATIONS
            PRINT #4, USING "#.######"; IR.READ.1#(2, CYCLE, READING);
            PRINT #4, "   ";
        NEXT READING
        PRINT #4,
NEXT CYCLE
PRINT #4,
PRINT #4, "46/44 RATIO....GAS #2"
PRINT #4,
FOR READING = 1 TO TOTAL.INTEGRATIONS
        PRINT #4, "readng "; RIGHT$(" " + STR$(READING), 2); "   ";
NEXT READING
PRINT #4, "Mass 44"
FOR CYCLE = 1 TO CYCLES
        FOR READING = 1 TO TOTAL.INTEGRATIONS
            PRINT #4, USING "#.######"; IR.READ.2#(2, CYCLE, READING);
            PRINT #4, "   ";
        NEXT READING
        PRINT #4, USING "##.###"; GAS#(2, CYCLE)
NEXT CYCLE
CLOSE #4
RETURN
```

```
save.setup:
    LOCATE 24, 1: PRINT SPC(79);
    LINE (140, 316)-(350, 355), , B
    BEEP
    LOCATE , 20: INPUT "File Name : ", setup.name$
    IF setup.name$ = "" THEN
        BEEP
        LOCATE , 20: PRINT SPC(20);
        GOTO save.setup
    END IF
    OPEN setup.name$ + ".stu" FOR OUTPUT AS #6
        FOR i = 1 TO 9
            PRINT #6, setup.line$(i); " "; setup.response$(i)
        NEXT i
    CLOSE #6
    file.exists = FALSE
    FOR i = 1 TO filecount
        IF RTRIM$(dirlist$(i - 1)) = UCASE$(setup.name$) THEN file.exists = TRUE
    NEXT i
    IF file.exists = FALSE THEN
        filecount = filecount + 1
        dirlist$(filecount - 1) = UCASE$(setup.name$)
    END IF
    setup.edited = FALSE
    RETURN

scaling.routine:

    ' The following routine will find our beginning voltage
    ' for each collector, and from this calculate the weight
    ' that each dot is worth for each collector.
    '
    IF AUTOSCALE = FALSE THEN
        VOLTS.PER.DOT.1# = FIXED.SCALE!
        VOLTS.PER.DOT.2# = FIXED.SCALE!
    ELSE
        VOLTS.PER.DOT.1# = ABS(MEAN.1#(1, 1) - MEAN.1#(2, 1)) / 62
        VOLTS.PER.DOT.2# = ABS(MEAN.2#(1, 1) - MEAN.2#(2, 1)) / 62
    END IF
    IF MEAN.1#(1, 1) < MEAN.1#(2, 1) THEN START.VOLTS.1# = MEAN.1#(1, 1) - (124 *
VOLTS.PER.DOT.1#) ELSE START.VOLTS.1# = MEAN.1#(2, 1) - (124 *
VOLTS.PER.DOT.1#)
    IF MEAN.2#(1, 1) < MEAN.2#(2, 1) THEN START.VOLTS.2# = MEAN.2#(1, 1) - (124 *
VOLTS.PER.DOT.2#) ELSE START.VOLTS.2# = MEAN.2#(2, 1) - (124 *
VOLTS.PER.DOT.2#)

    scale.factor = 1
    IF 2 * TOTAL.INTEGRATIONS * CYCLES > 230 THEN scale.factor = 2
    IF 2 * TOTAL.INTEGRATIONS * CYCLES > 460 THEN scale.factor = 3
    IF 2 * TOTAL.INTEGRATIONS * CYCLES > 690 THEN scale.factor = 4
    IF 2 * TOTAL.INTEGRATIONS * CYCLES > 920 THEN scale.factor = 5
    IF 2 * TOTAL.INTEGRATIONS * CYCLES > 1150 THEN scale.factor = 6
```

```
    '
    ' Now plot our first cycles worth of data.  From now on,
    ' the data will be plotted as it is collected
    '
    SCALE.CALCULATED = TRUE
    LAST.Y = 25
    FOR G = 1 TO 2
        FOR READING = 1 TO TOTAL.INTEGRATIONS
            NUMERATOR.1# = IR.READ.1#(G, CYCLE, READING)
            NUMERATOR.2# = IR.READ.2#(G, CYCLE, READING)
            GOSUB plot.data   'plot
        NEXT READING
    NEXT G
    RETURN

screen.dump:
    IF ioport$ = "LPT1:" THEN ioport% = 0:  ELSE ioport% = 1
    IF printer$ = "HPLaser" THEN
        printer% = 3: res% = 3: xm% = 1: ym% = 1
    ELSEIF printer$ = "PR-1012" THEN
        printer% = 0: res% = 2: xm% = 1: ym% = 1
    ELSEIF printer$ = "Epson LQ" THEN
        printer% = 1: res% = 5: xm% = 1: ym% = 1
    END IF
    CALL ScreenDump(printer%, ioport%, res%, xm%, ym%, 0, 0, prnerr%)
    RETURN



selection:
    a$ = INKEY$
    WHILE a$ <> CHR$(13) AND a$ <> CHR$(27)
        a$ = INKEY$
        IF LEN(a$) = 2 THEN
            IF RIGHT$(a$, 1) = CHR$(80) THEN
                item = item + 1
                IF item = maxitem + 1 THEN item = 1
                IF menu = 1 THEN GOSUB menu1 ELSE GOSUB menu2
            ELSEIF RIGHT$(a$, 1) = CHR$(72) THEN
                item = item - 1
                IF item = 0 THEN item = maxitem
                IF menu = 1 THEN GOSUB menu1 ELSE GOSUB menu2
            ELSEIF RIGHT$(a$, 1) = CHR$(59) THEN
                a$ = CHR$(13)
                delete.flag = TRUE
            END IF
        END IF
    WEND
    IF a$ = CHR$(27) THEN IF menu = 1 THEN item = 11       'quick quit
    RETURN
```

```
select.valve1:
    '     SELECT VALVE 1 (STANDARD GAS)
    IF STARTUP THEN
        GOSUB init.IEEE.488
        IR.OUT$ = "V2"
        GOSUB IR.output.routine
        STARTUP = FALSE
    END IF
    valve$ = "Valve 1 Selected"
    IR.OUT$ = "S1"
    GOSUB IR.output.routine      'OUTPUT COMMAND TO IR
    RETURN

select.valve2:
    '     SELECT VALVE 2 (SAMPLE GAS)
    IF STARTUP THEN
        GOSUB init.IEEE.488
        IR.OUT$ = "V2"
        GOSUB IR.output.routine
        STARTUP = FALSE
    END IF
    valve$ = "Valve 2 Selected"
    IR.OUT$ = "S2"
    GOSUB IR.output.routine
    RETURN

show.chauvenet:
    chauv$ = chauvenet$(VAL(setup.response$(line.number)))
    LOCATE line.number + 11, 50: PRINT chauv$;
    RETURN

show.directory:
    item = 1: maxitems = filecount
    GOSUB menu2
    LINE (365, 45)-(490, 65 + filecount * 15), 3, B
    LINE (367, 47)-(488, 63 + filecount * 15), 3, B
    COLOR 3
    LOCATE 6, 49: PRINT "F1 to delete"
    LOCATE 7 + maxitems, 48: PRINT "ENTER to load";
    COLOR 7
    RETURN

trigger.and.read.IR:
    FIRST = TRUE
    KEY(1) STOP
    DEV% = 4
    STAT% = INP(BAS% + 1)
    OUT (BAS% + 9), &HFD: GOSUB pause
    OUT (BAS% + 4), &H80
    OUT (BAS% + 5), 0
```

```
      WAIT (BAS% + 1), 2
      OUT (BAS% + 0), &H5F
      WAIT (BAS% + 1), 2
      OUT (BAS% + 0), &H3F
      OUT (BAS% + 4), &H40
      OUT (BAS% + 6), DEV%
      WAIT (BAS% + 1), 2
      OUT (BAS% + 0), (DEV% + &H20)
      OUT (BAS% + 4), &H80
      OUT (BAS% + 6), CARD%
      OUT (BAS% + 5), 0
      OUT (BAS% + 9), &HF6: GOSUB pause
      K$ = "B" + CHR$(13)
      FOR i = 1 TO 2
        a = ASC(MID$(K$, i, 1))
        WAIT (BAS% + 1), 2
        OUT (BAS% + 0), a
      NEXT i
      OUT (BAS% + 9), &HFD: GOSUB pause
PT1:    STAT% = INP(BAS% + 1)
      EOS% = 10
      OUT (BAS% + 9), &HFD: GOSUB pause
      OUT (BAS% + 4), &H80
      OUT (BAS% + 5), 0
      WAIT (BAS% + 1), 2
      OUT (BAS% + 0), &H5F
      WAIT (BAS% + 1), 2
      OUT (BAS% + 0), &H3F
      OUT (BAS% + 4), &H40
      OUT (BAS% + 6), DEV%
      WAIT (BAS% + 1), 2
      OUT (BAS% + 0), (DEV% + &H40)
      OUT (BAS% + 4), &H40
      OUT (BAS% + 6), CARD%
      IR.IN$ = ""
      OUT (BAS% + 5), 0
      OUT (BAS% + 9), &HF6: GOSUB pause
PT2:    WAIT (BAS% + 1), 1
      i = INP(BAS% + 0)
      IF i = EOS% GOTO PT3
      IR.IN$ = IR.IN$ + CHR$(i)
      GOTO PT2
PT3:    IF FIRST THEN NUMERATOR.1# = VAL(IR.IN$) / 100000!: FIRST = FALSE: IR.IN$
= "": GOTO PT1
      NUMERATOR.2# = VAL(IR.IN$) / 100000!
      OUT (BAS% + 0), &H5F
      OUT (BAS% + 9), &HFD: GOSUB pause
      KEY(1) ON
      RETURN
```

```
valve.select:
    LINE (375, 183)-(528, 216), , B
    LOCATE 12, 50
    PRINT "<ESC> to exit"
    LOCATE , 50: COLOR 12
    PRINT valve$: COLOR 7
    LOCATE , 50
    PRINT "<CR> to change"
    a$ = INKEY$
    WHILE a$ <> CHR$(27)
        IF a$ = CHR$(13) THEN
            IF INSTR(valve$, "1") <> 0 THEN
                GOSUB select.valve2
            ELSE
                GOSUB select.valve1
            END IF
            LOCATE 13, 50: COLOR 12: PRINT valve$: COLOR 7
        END IF
        a$ = INKEY$
    WEND
    FOR i = 1 TO 5: LOCATE 10 + i, 40: PRINT SPC(29); : NEXT i
    RETURN
```

```
'......................................................................

'                  PROGRAM "OXY"

'     written April-June 1991 by D.Macdonald
'     utilizes 'QuickWindows Advanced' routines from Software Interphase Inc.
'     QB /L QWADV.QLB path+OXY.BAS

'     Enter DEBUG mode by pressing F10 within first second!
'          This will eliminate pressure and temperature checks, door
'          status checks, and will reduce "no.stir.time" portion of
'          equilibrate step from 2hrs to 0.1 minute


'......................................................................
DECLARE SUB DELAY (delaytime$)
DECLARE SUB ACTIVATE (device%, state%)


     CLS
     REM $DYNAMIC     'allocate storage for arrays while program running
     DEFINT A-Z
     SCREEN 12
     CALL QWINIT(4)
     CALL QWSYSTEM(2, 1)               'enable error handling
     CALL BOLD14(1, qwerror): CALL ROMAN14(2, qwerror): CALL ITALICS14(3,
qwerror)
     CALL FONTSEL(2)
     PI = 3.141593
     CALL MSHOW
     CALL MSETY(0, 479)
     CALL MSETPOS(320, 240)
     radius = 10
     CONST F1.KEY = 15104   'Scan code for [F1] function key.
     CONST F10.KEY = 17408
     CONST MENU.CHECKED = 1  'Option-bit for menuoption routine to make menu
checkmarked.
     CONST ESC.KEY = 27      'Scan code for ESCAPE key.
     BASEADR% = &H340        'DAS-8 A/D converter board
     ignore.door = 0
     TRUE = 1: FALSE = 0
     DEBUG = FALSE
     samples.ready = FALSE   'set TRUE after samples equilibrated
     OK.to.proceed = TRUE    'set FALSE after

'-------------------- DIMENSION ARRAYS NEEDED ---------------------
     DIM item$(20)          'Individual menu items.
     DIM bar$(20)           'Strings for Menubar across the top of the screen.
     DIM kb(20)            'Keyboard Scancodes for Hot-Keys.
     DIM w1(20000)           'Int Array to hold screen under window.
```

```
    DIM w2(20000)        'Another window .
    DIM pm1(10000)        'Int array to hold screen beneath pulldown menu
    DIM pb1(2000)        'Array to hold screen under menu bar
    DIM state$(50), xpos(50), ypos(50)
    DIM sample$(16), selected(16), pressure$(16, 2)
    DIM relay(42)       'relay # corresponding to devices 1-->42

'********************************************************************************
'    ********* THIS IS THE MAIN MENU!!!! ************

    GOSUB INIT.RELAYS
    GOSUB PZ.SENSOR.CAL
    GOSUB READ.TIMING.FILE
    GOSUB SETUP.MENU
    GOSUB draw.sample.line
    time.delay = 1: GOSUB WAIT.A.WHILE    'time to press F10 for DEBUG mode
    DO
        GOSUB CHECK.MENU
        GOSUB CHECK.MOUSE
        GOSUB CHECK.PRESSURES
        GOSUB CHECK.TEMPERATURE
        IF RUNFLAG THEN GOSUB CHECK.PROCESS
    LOOP

'********************************************************************************
INIT.RELAYS:
    FOR i = 1 TO 42
        READ relay(i)
    NEXT i
    DATA 57,19,20,25,26,27,28,41,42,43,44,49,50,51,52,17,18,0,23,24,29
    DATA 30,31,32,45,46,47,48,53,54,55,56,21,22,0,33,36,58,35,34,59,38
    OPEN "com1:9600,n,8,2,ds,cs" FOR RANDOM AS #5
    'de-energize all relays
    FOR i = 17 TO 60
        PRINT #5, CHR$(2 * i - 2);
    NEXT i
    RETURN

'********************************************************************************
PZ.SENSOR.CAL:
    'voltage! to pressure calibration coefficients for thermocouple guage
    FOR i = 1 TO 8
        READ pcal!(i)
        READ vcal!(i)
    NEXT i
    DATA 2,1.695,1.1,1.841,0.59,2.158,0.245,2.95
    DATA 0.12,3.9,0.105,4.07,0.055,4.84,0.045,5.0
    RETURN

'********************************************************************************
READ.TIMING.FILE:
```

```
     'get delay times from file
     ON ERROR GOTO FILE.ERROR
     OPEN "TIMING.OXY" FOR INPUT AS #1
     INPUT #1, delay1$: delay1$ = RIGHT$("    " + delay1$, 4)
     INPUT #1, delay2$: delay2$ = RIGHT$("    " + delay2$, 4)
     INPUT #1, delay3$: delay3$ = RIGHT$("    " + delay3$, 4)
     INPUT #1, delay4$: delay4$ = RIGHT$("    " + delay4$, 4)
     INPUT #1, delay5$: delay5$ = RIGHT$("    " + delay5$, 4)
     INPUT #1, delay6$: delay6$ = RIGHT$("    " + delay6$, 4)
     INPUT #1, delay7$: delay7$ = RIGHT$("    " + delay7$, 4)
     INPUT #1, delay8$: delay8$ = RIGHT$("    " + delay8$, 4)
     INPUT #1, delay9$: delay9$ = RIGHT$("    " + delay9$, 4)
     INPUT #1, delay10$: delay10$ = RIGHT$("    " + delay10$, 4)
     CLOSE #1
T10:  ON ERROR GOTO 0
     RETURN


'******************************************************************************
CHECK.MENU:
     CALL MENUGET(pm1%(), menunumber%, optionnum%, flag%, kb%)
     IF flag = 2 THEN  '------ A key was pressed ------
          IF kb = F1.KEY THEN GOSUB HELP
          IF kb = F10.KEY THEN GOSUB TOGGLE.DEBUG
     END IF
     IF flag = 1 THEN
          SELECT CASE menunumber
               CASE 1
               IF optionnum = 1 THEN GOSUB SPECIFY.SAMPLES
               IF optionnum = 2 THEN GOSUB SPECIFY.TIMING
               CALL MENUCLOSE(pm1%)
               CASE 2
               IF optionnum = 1 THEN GOSUB START.PROCESS
               IF optionnum = 2 THEN GOSUB RESUME.PROCESS
               IF optionnum = 3 THEN GOSUB STOP.PROCESS
               CALL MENUCLOSE(pm1%)
               CASE 3
               GOSUB QUIT.PROGRAM
               CASE 4
               GOSUB SHOW.HELP
          END SELECT
     END IF
     flag = 0
     RETURN

'******************************************************************************
CHECK.MOUSE:
     CALL MOUSE(but, x, y)
'     LOCATE 29, 1: PRINT but, x, y;
     IF (but = 1) THEN GOSUB CHECK.FOR.HIT
'     LOCATE 29, 1. IF hit = 1 THEN PRINT "we have a hit!!";
```

```
    IF hit THEN
        IF samples.ready THEN GOSUB check.for.conflict
        IF OK.to.proceed THEN
            CALL MHIDE
            GOSUB switch.state
            CALL MOUSE(but, x, y)
            WHILE but = 1
                CALL MOUSE(but, x, y)
            WEND
            CALL MSHOW
        ELSE
            OK.to.proceed = TRUE
        END IF
        hit = 0
    END IF
    RETURN

'*****************************************************************************
CHECK.FOR.HIT:
    FOR i = 1 TO 42
        dist = SQR((y - ypos(i)) ^ 2 + (x - xpos(i)) ^ 2)
'       LOCATE 29, 1: PRINT i, dist, radius, hit; : a$ = INPUT$(1)
        IF dist <= radius THEN
            hit = 1: device = i
            RETURN
        END IF
    NEXT i
    RETURN

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

check.for.conflict:
    IF device = 36 AND state$(36) = "OFF" THEN
        FOR i = 19 TO 34
            IF state$(i) = "ON" THEN OK.to.proceed = FALSE
        NEXT i
    ELSEIF device >= 19 AND device <= 34 AND state$(device) = "OFF" THEN
        IF state$(36) = "ON" THEN OK.to.proceed = FALSE
        FOR i = 19 TO 34
            IF i <> device AND state$(i) = "ON" THEN OK.to.proceed = FALSE
        NEXT i
    END IF
    IF OK.to.proceed = FALSE THEN
        BEEP
        CALL WOPEN(10, 5, 35, 11, &H1, &H70, "", w2(), 1)
        CALL WCOLOR(1, &H71)
        CALL WPRINTA(1, 2, 0, &H71, "Are you sure it's OK")
        CALL WPRINTA(1, 2, 1, &H71, "to open this valve?")
        CALL DIALOGINIT(1, 2, 1)
        CALL PUSHBUTTON(1, 5, 2, 2, &H70, 0, "Yes", &H74)
        CALL PUSHBUTTON(2, 15, 2, 2, &H70, 0, "No", &H74)
```

```
        DO
            CALL DIALOGGET(dialog, flag, value)
            IF flag = 1 AND dialog = 1 THEN OK.to.proceed = TRUE
            IF flag = 1 AND dialog = 2 THEN OK.to.proceed = FALSE
            IF (dialog < 3) AND (flag = 0) AND (value = 13) THEN CALL
DIALOGSET(dialog + 1)
        LOOP WHILE flag = 0
    END IF
    CALL MHIDE
    CALL WCLOSE(1)
    CALL MSHOW
    RETURN

'****************************************************************************
TOGGLE.DEBUG:
    IF DEBUG = FALSE THEN DEBUG = TRUE ELSE DEBUG = FALSE
    RETURN
'****************************************************************************
CHECK.PRESSURES:
    IF DEBUG THEN RETURN
    pass = pass + 1   '   : LOCATE 5, 50: PRINT pass
    channel = 0       'PIEZOELECTRIC SENSOR
    OUT BASEADR% + 2, channel
    FOR i = 1 TO 10: NEXT i         'timing problem with A/D board
    OUT BASEADR% + 1, 0              'start conversion
    FOR i = 1 TO 10: NEXT i
ploop1: IF INP(BASEADR% + 2) >= 128 THEN GOTO ploop1
    XL% = INP(BASEADR%): XH% = INP(BASEADR% + 1)
    a2dcount! = 16 * XH% + XL% / 16
    voltage! = a2dcount! * (10 / 4096) - 5
'    LOCATE 22, 1: PRINT "pz : "; a2dcount!, voltage!
    press! = -573.73 + 301.43 * voltage!
    pzvoltsum! = pzvoltsum! + voltage!
    pzsum! = pzsum! + press!
    IF pass = 100 THEN
'       LOCATE 19, 60: PRINT "pz: "; : PRINT USING "#.###"; pzvoltsum! / pass
        LOCATE 19, 6
        IF (pzsum! / pass) > 2 THEN
            PRINT USING "####"; INT(pzsum! / pass)
        ELSE
            PRINT "...."
        END IF
        pzsum! = 0
        pzvoltsum! = 0
    END IF


    channel = 1       'THERMOCOUPLE SENSOR
    OUT BASEADR% + 2, channel
    FOR i = 1 TO 10: NEXT i
    OUT BASEADR% + 1, 0              'start conversion
```

```
    FOR i = 1 TO 10: NEXT i
ploop2:  IF INP(BASEADR% + 2) >= 128 THEN GOTO ploop2
    XL% = INP(BASEADR%): XH% = INP(BASEADR% + 1)
    a2dcount! = 16 * XH% + XL% / 16
    voltage! = a2dcount! * (10 / 4096) - 5
'    LOCATE 23, 1: PRINT "tc : "; a2dcount!; voltage!
    tcvoltsum! = tcvoltsum! + voltage!
    i = 1
    WHILE voltage! > vcal!(i)
        i = i + 1
    WEND
    press! = pcal!(i) + (vcal!(i) - voltage!) * (pcal!(i - 1) - pcal!(i)) / (vcal!(i) - vcal!(i - 1))
    tcsum! = tcsum! + press!
    IF pass = 100 THEN
        IF (tcvoltsum! / pass) < vcal!(1) THEN
            LOCATE 19, 18: PRINT "...."
        ELSE
'            LOCATE 20, 60:PRINT "tc: "; : PRINT USING "#.###"; tcvoltsum! / pass
            LOCATE 19, 18: PRINT USING "####"; INT(tcsum! / pass * 1000)
        END IF
        tcsum! = 0
        tcvoltsum! = 0
    END IF

    RETURN

'*************************************************************************
CHECK.TEMPERATURE:
    IF DEBUG THEN RETURN
    channel = 2        'TEMP sensor
    OUT BASEADR% + 2, channel
    FOR i = 1 TO 10: NEXT i
    OUT BASEADR% + 1, 0          'start conversion
    FOR i = 1 TO 10: NEXT i
tloop:  IF INP(BASEADR% + 2) >= 128 THEN GOTO tloop
    XL% = INP(BASEADR%): XH% = INP(BASEADR% + 1)
    a2dcount! = 16 * XH% + XL% / 16
    voltage! = a2dcount! * (10 / 4096) - 5
    ' FOR temp=APPROX. 22.4 DEG C, voltage!=4.10 V
'    LOCATE 24, 1: PRINT "temp : "; a2dcount!; voltage!
    temp = 22.4 - (4.1 - voltage!) / .035
    tempsum! = tempsum! + temp
    IF pass = 100 THEN
        LOCATE 21, 50: PRINT USING "+##"; tempsum! / pass
        tempsum! = 0
        pass = 0
    END IF
    RETURN
```

```
'*******************************************************************************
CHECK.PROCESS:
'        device #        description
'        1           switch : stirrer
'        2 - 17       valve  : sample (B0-->BF)
'        18          valve  : sample, ALL (B0+B1+...BF)
'        19- 34       valve  : reservoir (A0-->AF)
'        35          valve  : reservoir, ALL (A0+A1+...AF)
'        36          valve  : main line manifold
'        37          valve  : mass spec. line manifold
'        38          switch : Peltier cooler
'        39          valve  : CO2 tank
'        40          valve  : adjustable leak
'        41          valve  : vacuum pump
'        42          valve  : vent to atmosphere

    t# = TIMER
    IF t# < start.time# THEN
        'have passed through midnight when timer is reset
        elapsed.time# = 60# * 1440# - start.time# + t#
    ELSE
        elapsed.time# = t# - start.time#
    END IF
    IF elapsed.time# < delay.time# THEN        'gone thru midnight
        IF (delay.time# / (elapsed.time# + .0001) <= 154 / (time.count + 1)) AND task.number
< 11 THEN
            LINE (570, 245 - time.count - 1)-(580, 245 - time.count - 1)
            time.count = time.count + 1
        END IF
        RETURN
    END IF

    IF DEBUG = FALSE THEN
        GOSUB CHECK.DOORS
        no.stir.time# = 120#        'no stirring for last 2hrs of equilibration
    ELSE
        no.stir.time# = .1#
    END IF
    time.count = 0
    GOSUB REMOVE.TIMER
    IF subtask# = 1 OR selected(subtask# - 1) = 1 OR task.number = 7 THEN
        GOSUB ACTIVITY.PROMPT
        IF task.number < 11 THEN GOSUB START.TIMER
    END IF
    SELECT CASE task.number
        CASE 1
            'evacuate main manifold
            GOSUB LEAK.TEST.FLAG
            device = 1: state = 1: GOSUB ACTIVATE
```

```
        device = 41: state = 1: GOSUB ACTIVATE
        device = 36: state = 1: GOSUB ACTIVATE
        delaytime$ = delay1$: GOSUB DELAY
        task.number = task.number + 1
    CASE 2
        'evacuate reservoirs and sample tubes
        FOR i = 0 TO 15     'open reservoir valves
            IF selected(i) = 1 THEN device = i + 2: state = 1: GOSUB ACTIVATE
        NEXT i

        FOR i = 0 TO 15     'open sample valves
            IF selected(i) = 1 THEN device = i + 19: state = 1: GOSUB ACTIVATE
        NEXT i
        delaytime$ = delay2$: GOSUB DELAY
        task.number = task.number + 1
    CASE 3
        FOR i = 0 TO 15     'close reservoir valves
            IF selected(i) = 1 THEN device = i + 19: state = 0: GOSUB ACTIVATE
        NEXT i
        delaytime$ = delay3$: GOSUB DELAY
        task.number = task.number + 1
    CASE 4
        'do leak test
        IF selected(subtask# - 1) = 1 THEN
            device = 41: state = 0: GOSUB ACTIVATE
            GOSUB CHECK.TC.GUAGE
            IF tc.pressure! <> -999 THEN
                pressure$(subtask#, 1) = RIGHT$("    " + STR$(tc.pressure!), 4)
            ELSE
                pressure$(subtask#, 1) = "****"
            END IF
            device = subtask# + 18: state = 1: GOSUB ACTIVATE
            delaytime$ = delay10$: GOSUB DELAY
            GOSUB START.TIMER
            DO
                t# = TIMER
                elapsed.time# = t# - start.time#
                GOSUB CHECK.PRESSURES
                GOSUB CHECK.TEMPERATURE
                GOSUB CHECK.MENU
                GOSUB CHECK.MOUSE
                IF (delay.time# / (elapsed.time# + .0001) <= 154 / (time.count + 1))
AND task.number < 10 THEN
                    LINE (570, 245 - time.count - 1)-(580, 245 - time.count - 1)
                    time.count = time.count + 1
                END IF
            LOOP WHILE (t# - start.time#) < delay.time#
            GOSUB REMOVE.TIMER
            GOSUB CHECK.TC.GUAGE
            IF tc.pressure! <> -999 THEN
                pressure$(subtask#, 2) = RIGHT$("    " + STR$(tc.pressure!), 4)
```

```
            ELSE
                pressure$(subtask#, 2) = "*****"
            END IF
            device = subtask# + 18: state = 0: GOSUB ACTIVATE
            device = 41: state = 1: GOSUB ACTIVATE
            GOSUB START.TIMER
            delaytime$ = delay4$: GOSUB DELAY
        END IF
        subtask# = subtask# + 1
        IF subtask# > 16 THEN
            subtask# = 1
            task.number = task.number + 1
            BEEP
            GOSUB LEAK.REPORT
        END IF
    CASE 5
        'admit CO2 to reservoirs and tubes
        device = 41: state = 0: GOSUB ACTIVATE
        device = 39: state = 1: GOSUB ACTIVATE
        FOR i = 0 TO 15     'open reservoir valves
            IF selected(i) = 1 THEN device = i + 19: state = 1: GOSUB ACTIVATE
        NEXT i
        delaytime$ = delay5$: GOSUB DELAY
        task.number = task.number + 1
    CASE 6
        FOR i = 0 TO 15     'close reservoir valves
            IF selected(i) = 1 THEN device = i + 19: state = 0: GOSUB ACTIVATE
        NEXT i
        device = 39: state = 0: GOSUB ACTIVATE
        delaytime$ = delay6$: GOSUB DELAY
        task.number = task.number + 1
    CASE 7
        'close sample tubes and EQUILIBRATE
        FOR i = 0 TO 15     'close sample valves
            IF selected(i) = 1 THEN device = i + 2: state = 0: GOSUB ACTIVATE
        NEXT i
        IF VAL(LTRIM$(delay7$)) < no.stir.time# THEN
            delaytime$ = delay7$: GOSUB DELAY
            task.number = task.number + 1
        ELSE
            delay7a$ = STR$(VAL(LTRIM$(delay7$)) - no.stir.time#)
            delay7b$ = STR$(no.stir.time#)
            IF subtask# = 1 THEN
                delaytime$ = delay7a$
                subtask# = 2
                GOSUB DELAY
            ELSE
                device = 1: state = 0: GOSUB ACTIVATE
                delaytime$ = delay7b$: GOSUB DELAY
                subtask# = 1
                task.number = task.number + 1
```

```
            END IF
        END IF

    CASE 8
        'evacuate manifold and reservoirs; cooler on
        device = 38: state = 1: GOSUB ACTIVATE  'turn on Peltier cooler
        device = 41: state = 1: GOSUB ACTIVATE
        FOR i = 0 TO 15     'open reservoir valves
            IF selected(i) = 1 THEN device = i + 19: state = 1: GOSUB ACTIVATE
        NEXT i
        delaytime$ = delay8$: GOSUB DELAY
        task.number = task.number + 1
    CASE 9
        'admit CO2 to reservoirs
        FOR i = 0 TO 15     'close reservoir valves
            IF selected(i) = 1 THEN device = i + 19: state = 0: GOSUB ACTIVATE
        NEXT i
        FOR i = 0 TO 15     'open sample valves
            IF selected(i) = 1 THEN device = i + 2: state = 1: GOSUB ACTIVATE
        NEXT i
        delaytime$ = delay9$: GOSUB DELAY
        task.number = task.number + 1
    CASE 10
        'isolate reservoirs
        FOR i = 0 TO 15     'close sample valves
            IF selected(i) = 1 THEN device = i + 2: state = 0: GOSUB ACTIVATE
        NEXT i
        device = 1: state = 0: GOSUB ACTIVATE
        device = 36: state = 0: GOSUB ACTIVATE
        device = 41: state = 0: GOSUB ACTIVATE
        device = 37: state = 1: GOSUB ACTIVATE
        GOSUB REMOVE.TIMER
        BEEP
        CALL WOPEN(1, 2, 55, 5, &H1, &H70, "", w2(), 3)
        CALL WCOLOR(3, &H71)
        CALL WPRINTA(3, 1, 0, &H71, "Use mouse to introduce samples into Mass
Spec inlet.")
        CALL WPRINTA(3, 1, 1, &H71, "Press mouse button to continue.")
        WHILE flag = 0
            CALL MOUSE(flag, x, y)
        WEND
        WHILE flag <> 0
            CALL MOUSE(flag, x, y)
        WEND
        CALL WCLOSE(3)
        samples.ready = TRUE
        task.number = task.number + 1
    CASE 11
        task.number = 1
        RUNFLAG = 0
'        device = 38: state = 0: GOSUB ACTIVATE
```

```
'                   device = 37: state = 0: GOSUB ACTIVATE
        END SELECT
        RETURN

'*************************************************************************
START.TIMER:
        time.count = 0
        LINE (570, 90)-(580, 245), 7, B
        RETURN

'*************************************************************************
REMOVE.TIMER:
        FOR i = 0 TO 156
            LINE (570, 90 + i)-(580, 90 + i), 0
        NEXT i
        RETURN
'*************************************************************************
LEAK.TEST.FLAG:
        ' show results OR simply write to file
        SOUND 500, 1
        PAUSE.FLAG = FALSE  'default is don't stop after test
        CALL WOPEN(10, 5, 37, 10, &H1, &H70, "", w2(), 1)
        CALL WCOLOR(1, &H71)
        CALL WPRINTA(1, 2, 0, &H71, "Pause after leak test?")
        CALL DIALOGINIT(1, 2, 1)
        CALL PUSHBUTTON(1, 4, 1, 2, &H70, 0, "Yes", &H74)
        CALL PUSHBUTTON(2, 16, 1, 2, &H70, 0, "No", &H74)
        DO
            CALL DIALOGGET(dialog, flag, value)
            IF flag = 0 AND value = 27 THEN GOTO ESC40
            IF flag = 1 AND dialog = 1 THEN PAUSE.FLAG = TRUE
            IF flag = 1 AND dialog = 2 THEN PAUSE.FLAG = FALSE
        LOOP WHILE flag = 0
ESC40:  CALL MHIDE
        CALL WCLOSE(1)
        CALL MSHOW
        RETURN
'*************************************************************************


CHECK.DOORS:
        PRINT #5, CHR$(0);
        WHILE LOF(5) = 0
        WEND
        status$ = INPUT$(1, 5)
        linecount = 0
        IF (ASC(status$) AND &H3) <> &H3 THEN door(1) = 1 ELSE door(1) = 0
        IF (ASC(status$) AND &HC) <> &HC THEN door(2) = 1 ELSE door(2) = 0
        IF (ASC(status$) AND &H30) <> &H30 THEN door(3) = 1 ELSE door(3) = 0
        IF (ASC(status$) AND &HC0) <> &HC0 THEN door(4) = 1 ELSE door(4) = 0
        IF ASC(status$) <> 255 AND ignore.door = 0 THEN
            CALL WOPEN(10, 7, 47, 16, &H1, &H70, "", w2(), 1)
```

```
        CALL WCOLOR(1, &H71)
        IF door(1) = 1 THEN
            CALL WPRINTA(1, 2, 1 + linecount, &H71, "Door 'a' (samples  0- 3) is open")
            linecount = linecount + 1
        END IF
        IF door(2) = 1 THEN
            CALL WPRINTA(1, 2, 1 + linecount, &H71, "Door 'b' (samples  4- 7) is open")
            linecount = linecount + 1
        END IF
        IF door(3) = 1 THEN
            CALL WPRINTA(1, 2, 1 + linecount, &H71, "Door 'c' (samples  8-11) is open")
            linecount = linecount + 1
        END IF
        IF door(4) = 1 THEN
            CALL WPRINTA(1, 2, 1 + linecount, &H71, "Door 'd' (samples 12-15) is open")
        END IF
        CALL DIALOGINIT(1, 2, 1)
        CALL PUSHBUTTON(1, 2, 5, 2, &H70, 0, "OK", &H74)
        CALL PUSHBUTTON(2, 18, 5, 2, &H70, 0, "Ignore", &H74)
        DO
            CALL DIALOGGET(dialog, flag, value)
            IF flag = 0 AND value = 27 THEN GOTO ESC5
            IF flag = 1 AND dialog = 1 THEN GOTO ESC5
            IF flag = 1 AND dialog = 2 THEN
                ignore.door = 1
                GOTO ESC5
            END IF
            IF (dialog < 3) AND (flag = 0) AND (value = 13) THEN CALL
DIALOGSET(dialog + 1)
        LOOP
ESC5:     CALL MHIDE
        CALL WCLOSE(1)
        CALL MSHOW
        END IF
    RETURN

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

ACTIVITY.PROMPT:
    SELECT CASE task.number
        CASE 1
            prompt$ = "Step 1.  Evacuate main manifold.             "
        CASE 2
            prompt$ = "Step 2.  Evacuate reservoirs and sample tubes. "
        CASE 3
            prompt$ = "Step 3.  Close all reservoir valves.         "
        CASE 4
            prompt$ = "Step 4.  Leak test sample #" + HEX$(subtask# - 1) + ".        "
        CASE 5
            prompt$ = "Step 5.  Admit CO2 to reservoirs and tubes.   "
        CASE 6
            prompt$ = "Step 6.  Close all reservoir valves.         "
```

```
        CASE 7
            IF subtask# = 1 THEN
            prompt$ = "Step 7a. Isolate sample tubes and EQUILIBRATE. "
            ELSE
            prompt$ = "Step 7b. Continue to EQUILIBRATE; stirrer off."
            END IF
        CASE 8
            prompt$ = "Step 8.  Evacuate main manifold and reservoirs."
        CASE 9
            prompt$ = "Step 9.  Admit equilibrated CO2 to reservoirs. "
    END SELECT
    IF task.number = 1 THEN CALL WOPEN(8, 3, 58, 5, &H2, &H7, "", w1(), 4)
    IF task.number < 10 THEN CALL WPRINTA(4, 1, 0, &H7, prompt$)
    IF task.number = 10 THEN
        CALL MHIDE
        CALL WCLOSE(4)
        CALL MSHOW
    END IF
    RETURN

'*****************************************************************************
ACTIVATE:
    IF state = 1 THEN
        state$(device) = "ON"
        IF device <> 1 AND device <> 38 THEN
            colour = 4
        ELSE
            colour = 14
        END IF
    ELSE
        state$(device) = "OFF"
        IF device <> 1 AND device <> 38 THEN
            colour = 1
        ELSE
            colour = 3
        END IF
    END IF
    CALL MHIDE
    CIRCLE (xpos(device), ypos(device)), radius, colour
    PAINT (xpos(device), ypos(device)), colour, colour
    CALL MSHOW
    IF state$(device) = "ON" THEN
        PRINT #5, CHR$(2 * relay(device) - 1);
    ELSE
        PRINT #5, CHR$(2 * relay(device) - 2);
    END IF
    RETURN

'*****************************************************************************
DELAY:
    start.time# = TIMER
```

```
    delay.time# = 60 * VAL(LTRIM$(delaytime$))
    RETURN

'****************************************************************************
WAIT.A.WHILE:
    begin.time# = TIMER
    DO
        now.time# = TIMER
    LOOP WHILE (now.time# - begin.time#) < time.delay
    RETURN

'****************************************************************************
CHECK.TC.GUAGE:
    IF DEBUG THEN RETURN
    channel = 1        'THERMOCOUPLE SENSOR
    tcvoltsum! = 0
    OUT BASEADR% + 2, channel
    FOR pass = 1 TO 50
        OUT BASEADR% + 1, 0            'start conversion
        FOR i = 1 TO 10: NEXT i
ploop4:    IF INP(BASEADR% + 2) >= 128 THEN GOTO ploop4
        XL% = INP(BASEADR%): XH% = INP(BASEADR% + 1)
        a2dcount! = 16 * XH% + XL% / 16
        voltage! = a2dcount! * (10 / 4096) - 5
        tcvoltsum! = tcvoltsum! + voltage!
    NEXT pass
    pass = 0
    voltage! = tcvoltsum! / 50
    i = 1
    WHILE voltage! > vcal!(i)
        i = i + 1
    WEND
    IF voltage! < vcal(1) THEN
        tc.pressure! = -999
    ELSE
        tc.pressure! = pcal!(i) + (vcal!(i) - voltage!) * (pcal!(i - 1) - pcal!(i)) / (vcal!(i) - vcal!(i -
1))
        tc.pressure! = INT(tc.pressure! * 1000)
    END IF
    tcvoltsum! = 0
    RETURN


'****************************************************************************
LEAK.REPORT:
    IF PAUSE.FLAG THEN
      CALL WOPEN(10, 5, 43, 19, &H1, &H70, "Leak Test Report", w2(), 1)
      CALL WCOLOR(1, &H71)
      CALL WPRINTA(1, 2, 1, &H71, "# Pinit Ptest   # Pinit Ptest")
      FOR i = 1 TO 8
```

```
            CALL WPRINTA(1, 2, 2 + (i - 1), &H71, HEX$(i - 1) + SPACE$(1) + pressure$(i, 1)
+ SPACE$(1) + pressure$(i, 2))
            CALL WPRINTA(1, 18, 2 + (i - 1), &H71, HEX$(i + 7) + SPACE$(1) + pressure$(i +
8, 1) + SPACE$(1) + pressure$(i + 8, 2))
         NEXT i
         CALL DIALOGINIT(1, 2, 1)
         CALL PUSHBUTTON(1, 2, 10, 2, &H70, 0, "Proceed", &H74)
         CALL PUSHBUTTON(2, 18, 10, 2, &H70, 0, "Stop", &H74)
         DO
            CALL DIALOGGET(dialog, flag, value)
            IF flag = 0 AND value = 27 THEN GOTO ESC4
            IF flag = 1 AND dialog = 1 THEN GOTO ESC4
            IF flag = 1 AND dialog = 2 THEN
                 GOSUB STOP.PROCESS
                 GOTO ESC4
            END IF
            IF (dialog < 3) AND (flag = 0) AND (value = 13) THEN CALL DIALOGSET(dialog +
1)
         LOOP
      END IF
ESC4:   CALL MHIDE
      CALL WCLOSE(1)
      CALL MSHOW

      OPEN "leak.tst" FOR OUTPUT AS #6
      PRINT #6, "Leak test report for "; TIME$; SPC(2); DATE$
      FOR i = 1 TO 16
            PRINT #6, RIGHT$(" " + STR$(i), 2); SPC(5); pressure$(i, 1);
            PRINT #6, SPC(5); pressure$(i, 2)
      NEXT i
      CLOSE #6
      RETURN

'*****************************************************************************
SPECIFY.SAMPLES:
      'LOCATE 29, 1: PRINT "SPECIFY.SAMPLES ROUTINE   ";
      wstyle = 2
      CALL WOPEN(10, 5, 25, 19, &H1, &H70, "", w2(), 1)
      CALL WCOLOR(1, &H71)
      CALL DIALOGINIT(1, 17, 1)
      CALL VPICKBOX(1, 0, 0, 12, 8, 0, 16, VARPTR(sample$(0)), &H70, &H70, &H74, 251,
1)
      CALL PUSHBUTTON(17, 5, 10, 2, &H70, 0, "OK", &H74)
      DO
            CALL DIALOGGET(dialog, flag, value)
            IF flag = 1 AND dialog < 17 THEN selected(value - 1) = NOT selected(value - 1)
            IF flag = 0 AND value = 27 THEN GOTO ESC3
            IF flag = 1 AND dialog = 17 THEN GOTO ESC3
            IF (dialog < 18) AND (flag = 0) AND (value = 13) THEN CALL DIALOGSET(dialog
+ 1)
      LOOP
```

```
ESC3:   CALL WCLOSE(1)
        RETURN


'*****************************************************************************
SPECIFY.TIMING:
        'LOCATE 29, 1: PRINT "SPECIFY.TIMING ROUTINE    ";
        wstyle = 2


        'FOR i = 1 TO 10: savedelay$(i) = delay$(i): NEXT i
        CALL WOPEN(10, 5, 65, 21, &H2, &H70, "Enter Delay Times (minutes)", w2(), 2)
        CALL WCOLOR(2, &H71)
        CALL DIALOGINIT(2, 11, 1)
        CALL INPUTBOX(1, 5, 1, "Open rack to vacuum line             ", &H75, 266, 0,
".0123456789", delay1$, 0, 0)
        CALL INPUTBOX(2, 5, 2, "All reservoir/sample valves open     ", &H75, 266, 0,
".0123456789", delay2$, 0, 0)
        CALL INPUTBOX(3, 5, 3, "All reservoir valves closed          ", &H75, 266, 0,
".0123456789", delay3$, 0, 0)
        CALL INPUTBOX(4, 5, 4, "Test: sample line open               ", &H75, 266, 0,
".0123456789", delay10$, 0, 0)
        CALL INPUTBOX(5, 5, 5, "Test: pump vacuum line               ", &H75, 266, 0,
".0123456789", delay4$, 0, 0)
        CALL INPUTBOX(6, 5, 6, "Close s.valves: open CO2, all r.valves", &H75, 266, 0,
".0123456789", delay5$, 0, 0)
        CALL INPUTBOX(7, 5, 7, "Close all r.valves: open all s.valves ", &H75, 266, 0,
".0123456789", delay6$, 0, 0)
        CALL INPUTBOX(8, 5, 8, "Close all s.valves: EQUILIBRATE      ", &H75, 266, 0,
".0123456789", delay7$, 0, 0)
        CALL INPUTBOX(9, 5, 9, "Open vacuum line, all r.valves       ", &H75, 266, 0,
".0123456789", delay8$, 0, 0)
        CALL INPUTBOX(10, 5, 10, "Close all r.valves: open all s.valves ", &H75, 266, 0,
".0123456789", delay9$, 0, 0)
        CALL PUSHBUTTON(11, 5, 12, 2, &H71, 0, "OK", &H74)
        'CALL PUSHBUTTON(12, 20, 12, 2, &H71, 0, "Cancel", &H74)
        DO
            CALL DIALOGGET(dialog, flag, value)
            IF flag = 0 AND value = 27 THEN GOTO ESC2
            IF (dialog < 12) AND (flag = 0) AND (value = 13) THEN CALL DIALOGSET(dialog
+ 1)
        LOOP WHILE flag = 0
ESC2:   CALL WCLOSE(2)
        'IF dialog = 11 THEN FOR i = 1 TO 9: delay$(i) = savedelay$(i): NEXT i
        'save timing information
        OPEN "TIMING.OXY" FOR OUTPUT AS #1
        PRINT #1, delay1$
        PRINT #1, delay2$
        PRINT #1, delay3$
        PRINT #1, delay4$
        PRINT #1, delay5$
```

```
    PRINT #1, delay6$
    PRINT #1, delay7$
    PRINT #1, delay8$
    PRINT #1, delay9$
    PRINT #1, delay10$
    CLOSE #1
    RETURN

'*******************************************************************************
FILE.ERROR:
    'file not found
    delay1$ = "  .1"
    delay2$ = "  .1"
    delay3$ = "  .1"
    delay4$ = "  .1"
    delay5$ = "  .1"
    delay6$ = "  .1"
    delay7$ = "  .3"
    delay8$ = "  .1"
    delay9$ = "  .1"
    delay10$ = "  .1"
    RESUME T10
    RETURN


'*******************************************************************************
START.PROCESS:
'     LOCATE 29, 1: PRINT "START.PROCESS ROUTINE     ";
    IF STOPFLAG THEN
        GOSUB draw.sample.line
        STOPFLAG = 0
    END IF
    samples.ready = FALSE
    CALL WCLOSE(4)
    FOR i = 1 TO 16
        pressure$(i, 1) = ""
        pressure$(i, 2) = ""
    NEXT i
    delay.time# = 0: pass = 0
    FOR device = 1 TO 41          'turn off all valves and switches
        IF state$(device) = "ON" THEN
            PRINT #5, CHR$(2 * relay(device) - 2);
            GOSUB switch.state
        END IF
    NEXT device
    start.time# = TIMER
    task.number = 1: subtask# = 1
    RUNFLAG = 1
    RETURN

'*******************************************************************************
```

```
RESUME.PROCESS:
'      LOCATE 29, 1: PRINT "RESUME.PROCESS ROUTINE    ";
       t# = TIMER: start.time# = t# - elapsed.time#
       RUNFLAG = 1: STOPFLAG = 0
       RETURN

'**********************************************************************************
STOP.PROCESS:
'      LOCATE 29, 1: PRINT "STOP.PROCESS ROUTINE      ";
       t# = TIMER: elapsed.time# = t# - start.time#
       RUNFLAG = 0
       STOPFLAG = 1
       RETURN

'**********************************************************************************
QUIT.PROGRAM:
       'LOCATE 29, 1: PRINT "QUIT.PROGRAM ROUTINE      ";
       FOR device = 1 TO 41
            IF state$(device) = "ON" THEN PRINT #5, CHR$(2 * relay(device) - 2);
       NEXT device
       CLOSE
       SCREEN 0
       CLS
       END
       RETURN

'**********************************************************************************
SHOW.HELP:
'      LOCATE 29, 1: PRINT "SHOW.HELP ROUTINE         ";
       RETURN




       'Your application can act now upon the values of MenuNumber% and OptionNum%.
       'CALL MENUOFFA(pb1())   'Turn off the menu bar and restore screen.
       'END

'**********************************************************************************
HELP:
       'An actual help routine can go here.  This is just a window.
       CALL WOPEN(10, 7, 60, 13, &H74, &H75, "HELP WINDOW", w2(), 2)
       CALL WPRINT(2, " Help can be displayed for the current")
       CALL WPRINT(2, "~ Menu " + STR$(menunumber%) + " and its option " +
STR$(optionnum%) + ".")
       CALL WPRINT(2, "~~ Press ANY key to close window.")
       WHILE INKEY$ = "": WEND
       CALL WCLOSE(2)
       RETURN
```

```
'******************************************************************************
switch.state:
    IF device <> 1 AND device <> 38 THEN
        IF device <> 18 AND device <> 35 THEN
            IF state$(device) = "OFF" THEN
                state$(device) = "ON"
                PRINT #5, CHR$(2 * relay(device) - 1);
                circlecolour = 4
            ELSE
                state$(device) = "OFF"
                PRINT #5, CHR$(2 * relay(device) - 2);
                circlecolour = 1
            END IF
            CIRCLE (xpos(device), ypos(device)), radius, circlecolour
            PAINT (xpos(device), ypos(device)), circlecolour, circlecolour
        ELSE
            FOR i = (device - 16) TO (device - 1)
                IF x < 510 THEN
                    circlecolour = 4
                    state$(i) = "ON"
                    PRINT #5, CHR$(2 * relay(i) - 1);
                ELSE
                    circlecolour = 1
                    state$(i) = "OFF"
                    PRINT #5, CHR$(2 * relay(i) - 2);
                END IF
                CIRCLE (xpos(i), ypos(i)), radius, circlecolour
                PAINT (xpos(i), ypos(i)), circlecolour, circlecolour
            NEXT i
        END IF
    ELSE
        IF state$(device) = "OFF" THEN
            state$(device) = "ON"
            PRINT #5, CHR$(2 * relay(device) - 1);
            circlecolour = 14
        ELSE
            state$(device) = "OFF"
            PRINT #5, CHR$(2 * relay(device) - 2);
            circlecolour = 3
        END IF
        CIRCLE (xpos(device), ypos(device)), radius, circlecolour
        PAINT (xpos(device), ypos(device)), circlecolour, circlecolour
    END IF
    RETURN


'******************************************************************************
SETUP.MENU:

    FOR i = 0 TO 15
```

```
        sample$(i) = CHR$(251) + CHR$(32) + "sample #" + HEX$(i)'chr$(251)=checkmark
        selected(i) = 1
    NEXT i

'----- Now We Can Setup Our Menubar Across the Top of the Screen ------
    nummenus = 4
    barattr = &H4F
    barrev = &HF
    charen = &H24E     'Highlighted keys: Underline font set #2, attribute of 4F
    chardis = &H48
      'Option values (opts) have the following meanings:
      'Bit 0 set (Decimal 1) - String has hotkey position info, otherwise 1st char
      'Bit 1 set (2) - Bar String has menu title position information
      'Bit 2 set (4) - Place menubar inside window if wid<>0, otherwise on top.
      'Bit 3 set (8) - Don't close menu when exiting with flag=1.  Use MENUCLOSE.
    opts = 1 + 2 + 8

'---- Setup the menu bar array and the HotKeys for each menu ---
    bar$(0) = CHR$(1) + CHR$(0) + "Setup  "
'    bar$(1) = CHR$(1) + CHR$(0) + "Mode   "
    bar$(1) = CHR$(1) + CHR$(0) + "Run    "
    bar$(2) = CHR$(1) + CHR$(0) + "Quit   "
    bar$(3) = CHR$(4) + CHR$(255) + "F1=Help "
'               ^hotkey pos   ^menu title position (255=start from right)

'--- Call QuickWindows to Setup Menubar ----
    CALL MENUINIT(wid, nummenus, barattr, barrev, charen, chardis, VARPTR(bar$(0)),
opts)



'---- Read in each of the menu options & xfer into internal storage ---
'The *'s used below make it easier to specify the hotkeys in the data
'statements.  Of course the routine following the data statement parses the
'asterisks out and determines where the hotkeys are going.  If no asterisks
'are in the item name, then it is assumed that the first character of the
'item is the hot key.

    DATA 2,Samples,Timing
'    DATA 2,Automatic,Manual
    DATA 3,Start,Resume,S*top
    DATA 0
    DATA 0

    FOR MENU = 1 TO nummenus '--- For Each menu across the bar
        READ msize          '---The # of menu items (down) in each.
        IF msize <> 0 THEN
            FOR item = 1 TO msize   '--- For item in that menu.
                READ item$      '---This string is menu item itself.
                i = INSTR(item$, "*")
                IF i = 0 THEN
```

```
                        item$(item - 1) = CHR$(1) + item$        'hotkey for 1st letters
                    ELSE
                        item$(item - 1) = CHR$(i) + LEFT$(item$, i - 1) + MID$(item$, i + 1)
                    END IF
                NEXT item
                style = &H12: ahot = &HF: aen = &H7: adis = &H70: abdr = &H7
                CALL MENUSET(MENU, msize, style, abdr, aen, adis, ahot,
VARPTR(item$(0)))
        END IF
    NEXT MENU

'------ Turn on menu bar (Make It Visible) -------
    CALL MENUON


    flag = 0
    RETURN

'**************************************************************************
draw.sample.line:

    CALL MHIDE
    'stirrers
    vertpos = 100: colour = 3
    FOR j = 1 TO 16
        LOCATE INT(vertpos / 16), INT(j * 30 / 8) + 1
        IF j <> 17 THEN PRINT HEX$(j - 1);
        LINE (j * 30 - 10, vertpos - 5)-(j * 30 + 10, vertpos + 5), , B
        LINE (j * 30 + 10, vertpos)-(j * 30 + 20, vertpos)
    NEXT j
    CIRCLE (17 * 30, vertpos), radius, colour
    PAINT (17 * 30, vertpos), colour, colour

    'sample tubes
    vertpos = 125
    FOR j = 1 TO 16
        CIRCLE (j * 30, vertpos), 10, , 0, PI
        LINE (j * 30 - 10, vertpos)-(j * 30 - 10, vertpos + 30)
        LINE (j * 30 - 10, vertpos + 30)-(j * 30 + 10, vertpos + 30)
        LINE (j * 30 + 10, vertpos + 30)-(j * 30 + 10, vertpos)
        LINE (j * 30, vertpos + 30)-(j * 30, vertpos + 40)
    NEXT j

    'sample valves
    vertpos = 175: colour = 1
    FOR j = 1 TO 17
        CIRCLE (j * 30, vertpos), radius, colour
        PAINT (j * 30, vertpos), colour, colour
        IF j <> 17 THEN LINE (j * 30, vertpos + 10)-(j * 30, vertpos + 20)
        IF j = 17 THEN
            CIRCLE (j * 30, vertpos), radius, 4, -PI / 2, -3 * PI / 2
```

```
            PAINT (j * 30 - 1, vertpos), 4, 4
        END IF
NEXT j

'reservoirs
vertpos = 195
FOR j = 1 TO 16
        LINE (j * 30 - 10, vertpos)-(j * 30 + 10, vertpos + 20), , B
        LINE (j * 30, vertpos + 20)-(j * 30, vertpos + 30)
NEXT j

'reservoir valves
vertpos = 235
FOR j = 1 TO 17
        CIRCLE (j * 30, vertpos), radius, colour
        PAINT (j * 30, vertpos), colour, colour
        IF j <> 17 THEN LINE (j * 30, vertpos + 10)-(j * 30, vertpos + 20)
        IF j = 17 THEN
            CIRCLE (j * 30, vertpos), radius, 4, -PI / 2, -3 * PI / 2
            PAINT (j * 30 - 1, vertpos), 4, 4
        END IF
NEXT j
LINE (30, vertpos + 20)-(480, vertpos + 20)

vertpos = 255
LINE (7.5 * 30, vertpos)-(7.5 * 30, vertpos + 20)
LINE (10.5 * 30, vertpos)-(10.5 * 30, vertpos + 20)

'I/O valves, cooler switch
vertpos = 285
FOR j = 1 TO 3
        IF j = 3 THEN colour = 3
        CIRCLE (7.5 * 30 + (j - 1) * 90, vertpos), radius, colour
        PAINT (7.5 * 30 + (j - 1) * 90, vertpos), colour, colour
NEXT j

LOCATE 21, 46: PRINT "PC:     "; CHR$(248); "C    MS inlet"
vertpos = 320
LINE (315, 295)-(315, 329)
LINE (315, 329)-(350, 329)
LINE (405, 295)-(405, 314)
LINE (350, 314)-(445, 344), , B
LINE (445, 329)-(460, 329)

'MS inlet box
' LINE (460, 314)-(545, 344), , B
LINE (460, 314)-(460, 344)
LINE (460, 314)-(540, 314)
LINE (460, 344)-(540, 344)
LINE (540, 314)-(545, 329)
LINE (540, 344)-(545, 329)
```

```
'vent to atmos., CO2 tank, vac. pump valves
vertpos = 350: colour = 1
FOR j = 1 TO 4
      CIRCLE (7.5 * 30 - (j - 1) * 60, vertpos), radius, colour
      PAINT (7.5 * 30 - (j - 1) * 60, vertpos), colour, colour
NEXT j

LOCATE 19, 2: PRINT "PG:          TG: "
LINE (0, 280)-(84, 310), , B
LINE (93, 280)-(173, 310), , B
LINE (45, 310)-(45, 340)
LINE (133, 310)-(133, 325)

LINE (225, 295)-(225, 340)
LINE (45, 325)-(225, 325)
LINE (165, 325)-(165, 340)
LINE (105, 325)-(105, 340)
LINE (105, 360)-(105, 380)
LINE (45, 360)-(45, 370)
LINE (25, 370)-(45, 370)
LINE (165, 360)-(165, 370)
'vent to atmos. symbol
LINE (20, 365)-(25, 370)
LINE (20, 375)-(25, 370)
' adjustable leak symbol
LINE (180, 375)-(193, 362)
LINE (186, 365)-(193, 362)
LINE (193, 362)-(192, 368)

LINE (165, 370)-(225, 370)
LINE (225, 360)-(225, 380)

LOCATE 25, 13: PRINT "CO2          pump"
vertpos = 390
LINE (7.5 * 30 - 25, vertpos - 10)-(7.5 * 30 + 25, vertpos + 15), , B
LINE (3.5 * 30 - 25, vertpos - 10)-(3.5 * 30 + 25, vertpos + 15), , B

'legend
LINE (400, 380)-(530, 460), , B
LOCATE 25, 60: PRINT "off on"
LOCATE 28, 52: PRINT "switch"
LOCATE 26, 52: PRINT "valve"
FOR j = 1 TO 2
      vertpos = 410 + (j - 1) * 30: colour = 2 * j - 1
      CIRCLE (480, vertpos), radius, colour
      PAINT (480, vertpos), colour, colour
      CIRCLE (510, vertpos), radius, colour + 3 + (j - 1) * 8
      PAINT (510, vertpos), colour + 3 + (j - 1) * 8, colour + 3 + (j - 1) * 8
NEXT j
```

```
    'input coordinates of 42 valves and switches
    xpos(1) = 510: ypos(1) = 100    'stirrer switch
    FOR i = 2 TO 18                 'sample valves 0-->F
        xpos(i) = 30 + (i - 2) * 30: ypos(i) = 175
    NEXT i
    FOR i = 19 TO 35                'reservoir valves 0-->F
        xpos(i) = 30 + (i - 19) * 30: ypos(i) = 235
    NEXT i
    FOR i = 36 TO 38                'v.mnl, v.msl, s.cooler
        xpos(i) = 225 + (i - 36) * 90: ypos(i) = 285
    NEXT i
    FOR i = 39 TO 41                'v.co2, v.lek, v.vac
        xpos(i) = 105 + (i - 39) * 60: ypos(i) = 350
    NEXT i
    xpos(42) = 45: ypos(42) = 350

    FOR device = 1 TO 42
        state$(device) = "OFF"
    NEXT device
    CALL MSHOW
    RETURN      ' END draw.sample.line
'*****************************************************************************

REM $STATIC
SUB DELAY (delaytime$)
    start.time# = TIMER
    delay.time# = VAL(delaytime$)
END SUB
```