

NOTES TECHNIQUES
SCIENCES DE LA MER
OCÉANOGRAPHIE PHYSIQUE

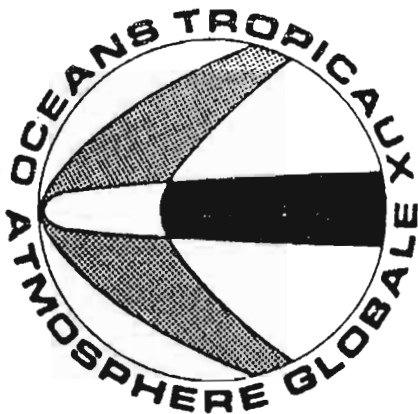
N° 7

1992

Installation et utilisation d'un thermosalinographe
à bord d'un navire marchand

Jacques GRELET
Bruno BUISSON
Christian HENIN

Groupe SURTROPAC



FINANCEMENT CORDET

L'INSTITUT FRANÇAIS DE RECHERCHE SCIENTIFIQUE
POUR LE DÉVELOPPEMENT EN COOPÉRATION

CENTRE DE NOUMÉA

ORSTOM

NOTES TECHNIQUES
SCIENCES DE LA MER
OCEANOGRAPHIE PHYSIQUE

N° 7

1992

**Installation et utilisation d'un thermosalinographe
à bord d'un navire marchand**

* Jacques GRELET
** Bruno BUISSON
* Christian HENIN

* *Groupe Surtropac*
** Laboratoire d'Informatique

FINANCEMENT CORDET

ORSTOM

L'INSTITUT FRANÇAIS DE RECHERCHE SCIENTIFIQUE
POUR LE DÉVELOPPEMENT EN COOPÉRATION

CENTRE DE NOUMÉA

© ORSTOM, Nouméa, 1992

/Grelet, J.
/Buisson, B.
/Hénin, C.

Installation et utilisation d'un thermosalinographe à bord d'un navire marchand
Nouméa : ORSTOM. Mai 1992, 99 p.
Notes tech. : Sci. Mer : Océanogr. phys. ; 7

Ø32TECHNO

OCEANOGRAPHIE PHYSIQUE ; TECHNOLOGIE ; SALINITE DE SURFACE ; TEMPERATURE DE
SURFACE ; CONDUCTIVITE ; ACQUISITION DE DONNEES ; THERMOSALINOGRAPHE /
NOUVELLE CALEDONIE

Imprimé par le Centre ORSTOM
de Nouméa
Mai 1992



TABLE DES MATIERES

| | |
|---|----|
| 1. INTRODUCTION | 9 |
| 2. DESCRIPTION DU SYSTEME D'ACQUISITION | 11 |
| 2.1. Thermosalinographe Sea-Bird SBE21..... | 11 |
| 2.1.1. Caractéristiques | 11 |
| 2.1.2. Description du système d'acquisition..... | 12 |
| 2.1.3. Calibration | 13 |
| 2.1.3.1. Calcul de la température | 13 |
| 2.1.3.2. Calcul de la conductivité..... | 13 |
| 2.1.3.3. Calcul de la salinité..... | 14 |
| 2.1.4. Interface..... | 15 |
| 2.2. Navigateurs par satellite..... | 16 |
| 2.2.1. Magnavox MX-1107..... | 16 |
| 2.2.2. Pronav GPS-100..... | 16 |
| 2.2.3. Centrale NALNO..... | 16 |
| 2.3. Navigation manuelle | 17 |
| 3. CONNEXION DES APPAREILS | 19 |
| 3.1. Liaison série: la norme RS232..... | 19 |
| 3.2. Multiplexeur de lignes RS232 | 20 |
| 3.2.1. Principe..... | 20 |
| 3.2.2. Description technique et principe de fonctionnement..... | 20 |
| 3.2.3. Utilisation | 20 |
| 3.2.4. Avantages et inconvénients | 20 |
| 4. LES MODULES LOGICIELS | 23 |
| 4.1. Les modes d'appel du programme..... | 23 |
| 4.2. Le module d'acquisition..... | 23 |
| 4.3. Le module de configuration du logiciel..... | 23 |
| 4.4. Le module de configuration du thermosalinographe..... | 23 |
| 4.5. Le mode terminal | 23 |
| 5. MISE EN OEUVRE DU LOGICIEL | 25 |
| 5.1. Philosophie de développement..... | 25 |
| 5.1.1. Une programmation multitâches | 25 |
| 5.1.2. Une programmation par objets..... | 26 |
| 5.1.3. Une programmation 'par couches successives' | 27 |
| 5.2. Les bibliothèques logicielles..... | 28 |
| 5.2.1. Une gestion de ports série sous interruptions..... | 28 |
| 5.2.1.1. Principes généraux..... | 28 |
| 5.2.1.2. Les interruptions matérielles..... | 28 |
| 5.2.1.3. Cas des ports série | 29 |
| 5.2.1.4. Gestion du multiplexeur..... | 29 |
| 5.2.1.5. Les fonctions d'interface..... | 30 |
| 5.2.1.6. Le niveau TRAME..... | 32 |
| 5.2.2. Une gestion de compteurs sous interruptions..... | 34 |
| 5.2.2.1. Les besoins..... | 34 |
| 5.2.2.2. Principe..... | 34 |
| 5.2.2.3. Les fonctions d'interface..... | 34 |

| | | |
|------------|---|-----------|
| 5.2.3. | L'horloge temps réel des IBM-AT | 36 |
| 5.2.3.1. | Description..... | 36 |
| 5.2.3.2. | Le rôle dans le programme..... | 36 |
| 5.2.3.3. | Les fonctions d'interface..... | 37 |
| 5.3. | Un ensemble de tâches communicantes | 37 |
| 5.3.1. | L'organisation du programme..... | 37 |
| 5.3.2. | Les tâches | 38 |
| 5.3.3. | Les signaux..... | 38 |
| 5.3.4. | Fonctionnement d'une tâche..... | 39 |
| 5.3.5. | Manipulation des tâches | 40 |
| 5.3.6. | Les différents états d'une tâche..... | 41 |
| 5.3.7. | Description d'une tâche d'acquisition..... | 42 |
| 5.3.8. | Description d'une tâche de stockage de données | 43 |
| 5.3.9. | Les outils de mise au point..... | 44 |
| 5.3.9.1. | Le mode trace..... | 44 |
| 5.3.9.2. | Le mode simulation..... | 44 |
| 6. | L'UTILISATION DE SF DANS LE PROGRAMME | 45 |
| 6.1. | Les principes généraux de SF | 45 |
| 6.2. | La bibliothèque de gestion d'écrans..... | 46 |
| 6.2.1. | Description | 46 |
| 6.2.2. | Les fonctions d'interface..... | 46 |
| 6.3. | La bibliothèque de gestion de fichiers | 47 |
| 6.3.1. | Description | 47 |
| 6.3.2. | Les fonctions d'interface..... | 48 |
| 6.4. | SF dans le programme THERMO | 50 |
| 6.4.1. | Configuration | 50 |
| 6.4.2. | Saisie de la position manuelle | 50 |
| 6.4.3. | Présentation des données..... | 50 |
| 7. | UTILISATION DU PROGRAMME THERMO..... | 53 |
| 7.1. | Principes | 53 |
| 7.2. | Mise en route du thermosalinographe..... | 54 |
| 7.3. | Utilisation des écrans dans le programme..... | 55 |
| 7.4. | Utilisation du système..... | 56 |
| 7.5. | Options de la ligne de commande..... | 59 |
| 7.6. | Les fichiers présents sur la disquette programme..... | 63 |
| 8. | RESULTATS..... | 65 |
| 9. | CONCLUSION | 69 |
| 10. | BIBLIOGRAPHIE..... | 71 |
| 11. | ANNEXE A: CARACTERISTIQUES DES APPAREILS..... | 73 |
| 11.1. | Thermosalinographe SBE21 | 73 |
| 11.1.1. | Description technique..... | 73 |
| 11.1.2. | Commandes | 73 |
| 11.2. | Magnavox MX-1107 | 74 |
| 11.2.1. | Description technique..... | 74 |
| 11.2.2. | Commandes | 74 |
| 11.3. | Pronav GPS-100..... | 74 |
| 11.3.1. | Description technique..... | 74 |
| 11.3.2. | Commandes | 75 |

| | |
|--|-----------|
| 11.4. Centrale NALNO..... | 76 |
| 11.4.1. Description technique..... | 76 |
| 11.4.2. Commandes | 76 |
| 12. ANNEXE B: REALISATION TECHNIQUE..... | 77 |
| 12.1. Synoptique du système d'acquisition | 77 |
| 12.2. Synoptique du système d'acquisition avec multiplexeur..... | 78 |
| 12.3. Réalisation d'un multiplexeur de voies série..... | 79 |
| 12.3.1. Câblage..... | 79 |
| 12.3.2. Schéma électronique..... | 80 |
| 12.3.3. Implantation des composants..... | 81 |
| 12.3.4. Circuit imprimé..... | 82 |
| 12.3.5. Nomenclature | 83 |
| 12.4. Préparation d'un thermosalinographe..... | 84 |
| 12.4.1. Support de fixation..... | 84 |
| 12.4.2. Assemblage des accouplements..... | 84 |
| 12.4.3. Check-list d'installation..... | 84 |
| 12.4.4. Schéma d'implantation à bord d'un navire..... | 86 |
| 12.4.5. Installation sur site..... | 87 |
| 12.4.6. Description de l'ensemble moteur/pompe..... | 88 |
| 13. ANNEXE C | 89 |
| 13.1. Structure du descripteur de fichier config.dsk..... | 89 |
| 13.2. Exemple de fichier config.cfg..... | 89 |
| 13.3. Structure du descripteur de fichier constant.dsk..... | 90 |
| 13.4. Exemple de fichier constant.cfg..... | 90 |
| 13.5. Format du fichier de données | 90 |
| 13.6. Calendrier des jours juliens..... | 91 |
| 14. ANNEXE D | 93 |
| 14.1. Coût du système..... | 93 |
| 14.2. Adresses des fournisseurs..... | 94 |
| 15. GLOSSAIRE | 95 |
| 16. INDEX..... | 97 |

RESUME

La distribution de température et de la salinité dans l'Océan et sa variabilité saisonnière et interannuelle est très importante pour comprendre le rôle de l'Océan dans le cycle climatique de la planète. Le groupe SURTROPAC (SURveillance TROPicale du PACifique) du Centre ORSTOM de Nouméa réalise depuis 1969 des mesures de température et des prélèvements de salinité au seau météorologique sur des navires de commerce. Il apparaît de plus en plus nécessaire, si l'on veut comprendre les mécanismes mis en jeu dans le couplage Océan/Atmosphère, de réaliser des mesures plus précises et plus nombreuses.

Le groupe SURTROPAC a obtenu en 1990 un financement CORDET (Commission de Coordination de la Recherche dans les Départements et Territoires d'Outre Mer) pour étudier la faisabilité d'une automatisation des mesures de température et salinité de surface. Cette notice décrit en détail la réalisation et l'utilisation du système de mesures embarqué sur les navires marchands, ainsi que les diverses techniques, informatiques et électroniques, employées dans cette réalisation. Ce système utilise un thermosalinographe de marque Sea-Bird, il a été développé et réalisé dans les laboratoires du groupe SURTROPAC et d'Informatique du centre ORSTOM de Nouméa (Nouvelle Calédonie).

ABSTRACT

The distribution of temperature and salt in the world oceans and their seasonal and inter-annual variability are important in understanding the role of the ocean in the Earth's climate. Bucket sea-surface temperature and salinity measurements are being obtained since 1969 through a Ship of Opportunity Program (SOP) managed by the SURTROPAC group (SURvey TROPical du PACific), of the ORSTOM center in Noumea, New Caledonia. However, a better understanding of the Ocean/Atmosphere coupling call for more accurate and numerous SST (Sea Surface Temperature) and SSS (Sea Surface Salinity) measurements in the tropical ocean.

Thanks to a CORDET (Commission de Coordination de la Recherche dans les Départements et Territoires d'Outre Mer) founding, a feasibility study has been undertaken by SURTROPAC in order to automatically measure SST and SSS on ship.

Our goal is to provide information about technical aspects related to SST and SSS monitoring. The used system is based on a Sea-Bird SBE21 thermosalinograph.

1. INTRODUCTION

Le principal objectif du programme international de recherche TOGA (Tropical Ocean Global Atmosphere) est de définir les origines des fluctuations du climat de notre planète à l'échelle de quelques mois à quelques années. L'océan par ses interactions avec l'atmosphère exerce une grande influence sur le climat. Pour arriver à ce but il est nécessaire de :

- comprendre et prévoir le système couplé océan-atmosphère.
- modéliser le système dans le but de prévoir ses fluctuations.
- améliorer les réseaux d'observations ainsi que la transmission des mesures en temps réel.

Le groupe SURTROPAC (SURveillance TROPicale du PACifique) du Centre ORSTOM de Nouméa est impliqué dans le programme TOGA depuis 1985 et y participe activement grâce, entre autres, à un réseau d'observations par navires de commerce. Depuis 1969, soit plus de vingt ans, des observations de température de surface (SST) et de salinité de surface (SSS) sont assurées, bénévolement, par les officiers des navires sillonnant le Pacifique. Ces mesures sont effectuées toutes les 6 heures à l'aide d'unseau météorologique.

Pour arriver aux objectifs de TOGA, il est apparu indispensable de moderniser ce réseau d'observations par des navires de commerce dans le Pacifique grâce à l'installation de thermosalinographes sur des bateaux traversant la zone tropicale régulièrement.

Sous l'impulsion de C.Henin, le groupe SURTROPAC a fait une demande de financement CORDET (Commission de Coordination de la Recherche dans les Départements et Territoires d'Outre Mer) début 1989 pour étudier la faisabilité d'une automatisation des mesures de température et salinité de surface.

Ce projet a été financé fin 1990, mais le développement d'un système d'acquisition automatique avait été entrepris avant cette date. C.Hamon, alors VAT à l'atelier d'informatique de Nouméa, a commencé à travailler sur un premier prototype avec une sonde STD de marque Applied System (disponible au laboratoire) et un récepteur de positionnement par satellite Transit de marque Magnavox. Ce système a été utilisé lors de la campagne SURTROPAC13 sur le N/O Le Suroît en juillet 1990.

L'étude entreprise dans le cadre du financement CORDET a démontré la faisabilité d'un tel système d'acquisition automatique. L'analyse des résultats a confirmé l'intérêt d'utiliser un thermosalinographe automatique. Notre choix s'est porté sur un appareil de marque Sea-Bird, modèle SBE21. Il fut alors décidé de mettre sur pied un logiciel pouvant servir de plate-forme de développement commune pour toutes les applications d'acquisition automatique de données en temps réel.

Le programme que nous décrivons ici est l'application de cette plate-forme à l'acquisition des données de SST et SSS. Une grande partie de ce qui est décrit dans ce document s'applique à d'autres applications réalisées à ce jour: magnétométrie des fonds sous-marins pour l'UR de géophysique, analyse du carbone dissous, programme ECOA (Grelet al, 1992) .

2. DESCRIPTION DU SYSTEME D'ACQUISITION

2.1. Thermosalinographe Sea-Bird SBE21

Le thermosalinographe Sea-Bird (modèle SBE21), est vendu, comme tous les appareils de la gamme Sea-Bird, avec un ensemble de logiciels d'exploitation appelé SEASOFT. Le logiciel SEASAVE permet de traiter indifféremment les données issues d'une sonde CTD (SBE09 ou SBE911 plus), d'une sonde Sea Cat (SBE16), ou d'un thermosalinographe (SBE21), après conversion éventuelle. A l'origine, le thermosalinographe envoyait les données via une liaison série et les stockait en mémoire vive (RAM) sous la forme d'un couple température-conductivité. Deux méthodes de traitement étaient alors possibles:

- 1) Les données étaient stockées en aveugle dans la RAM de l'appareil à un rythme choisi par l'utilisateur (15226 cycles maximum avec 64 Koctets de RAM). Les données étaient ensuite transférées sur un micro-ordinateur via le port série avec un logiciel appelé SCTERM et mises au format de SEASAVE avec un deuxième logiciel appelé SCSOFT.
- 2) L'acquisition se faisait comme précédemment, mais les données étaient acquises sur un micro-ordinateur en temps réel via le port série et traitées avec le logiciel SEASAVE.

Le logiciel fourni par Sea-Bird ne permet pas l'acquisition des données de navigation. Cette méthode nécessite un important travail de dépouillement au laboratoire. Dans le cadre d'une utilisation sur des bateaux marchands nous avons été amené à développer un logiciel spécifique permettant l'acquisition automatique de la position du navire.

Une première version a été installée sur le navire marchand TA Mariner en Novembre 1990. Il est très vite apparu que le mode de fonctionnement du thermosalinographe n'était pas satisfaisant pour une utilisation sur un navire marchand où les interventions humaines doivent être réduites au minimum. En effet une interruption de la tension d'alimentation, chose fréquente sur les navires de commerce, suffisait à stopper l'acquisition en continu des données, et nécessitait une intervention manuelle de l'officier, pour re-configurer le thermosalinographe et relancer l'acquisition. Nous avons donc suggéré à Sea Bird d'ajouter une commande supplémentaire permettant à l'ordinateur d'interroger l'appareil à la demande, sans stockage dans la RAM, et quelque soit le mode d'acquisition (actif ou repos). Une nouvelle version de la ROM contenant le logiciel nous a donc été fournie pour notre premier appareil, complété par la commande **TS** (Take Sample).

Cette version est maintenant livrée en standard sur tous les thermosalinographes et nous donne entière satisfaction. En conséquence, le système peut être relancé sans intervention manuelle et l'acquisition n'est plus limitée par la taille de la RAM du thermosalinographe.

2.1.1. Caractéristiques

Le système dans sa version actuelle (V4.0), fonctionne avec 3 types de récepteur de positionnement par satellite:

- Le Magnavox (modèle MX 1107) est utilisé sur le N/O Alis.
- NALNO est la sortie de la centrale de navigation du N/O Le Noroît utilisée par l'ADCP (Acoustic Doppler Current Profileur).
- NMEA-0183 est un format de données disponible actuellement sur la majorité des récepteurs GPS du commerce.

La position peut être saisie manuellement en cas d'absence ou de panne de l'appareil de navigation. Le programme de traitement se charge ensuite de recalculer la route par interpolation linéaire entre les points de saisie. La configuration du système est enregistrée sur disque et aisément paramétrable. Un mode terminal permet de dialoguer avec le périphérique de son choix. Le logiciel peut utiliser un multiplexeur de voies série RS232, permettant d'interfacer jusqu'à quatre équipements avec un micro-ordinateur portable ne possédant qu'un port série (cf § 11.3).

2.1.2. Description du système d'acquisition

Le système d'acquisition complet installé sur les navires marchands est constitué des éléments suivants:

- Un thermosalinographe Sea-Bird modèle SBE21 comprenant l'unité de mesure, la boîte de jonction et le câble d'alimentation.
- Un récepteur Pronav GPS100 avec son antenne extérieure et une alimentation externe 12 Volts.
- Un micro-ordinateur portable de type XT avec deux lecteurs de disquettes 3 1/2 pouces .
- Un multiplexeur de voies série RS232.
- Un ensemble moteur électrique + pompe pour l'alimentation en eau de mer du thermosalinographe, si son installation à la machine n'est pas possible sur le circuit interne de refroidissement du navire.

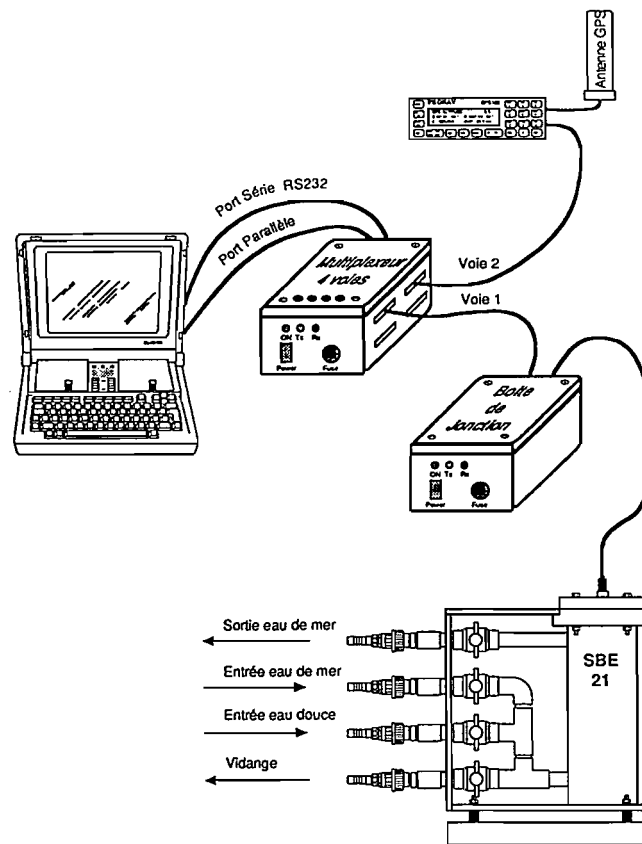


Fig 1. Système d'acquisition avec un port série et multiplexeur.

2.1.3. Calibration

Le thermosalinographe est livré avec des coefficients d'étalonnage pour le capteur de température et le capteur de conductivité. Chaque capteur transmet ses informations sous forme d'une fréquence. La grandeur mesurée est calculée à partir de la fréquence et des coefficients du capteur, grâce à un algorithme approprié.

Les valeurs de fréquences envoyées par le thermosalinographe doivent être corrigées avant d'être utilisées dans l'algorithme. Cette méthode présente l'avantage d'optimiser l'utilisation de l'espace disponible dans la RAM lors d'une acquisition in-situ.

Exemple: La commande TS (Take Sample) permet de réaliser une acquisition et transmet le résultat sans stocker les données dans la RAM

TS(CR) Commande Take Sample suivie d'un retour chariot-fin de ligne (code ASCII 13 et 10).

#ttttcccc où
caractère de début de trame.
tttt valeur de température non corrigée en fréquence.
cccc valeur de conductivité non corrigée en fréquence.

Prenons comme exemple ttttcccc = A80603DA en hexadécimal.

2.1.3.1. Calcul de la température

L'envoi de la trame est suivi d'un retour chariot-fin de ligne.

La valeur corrigée de température = $tttt / 19 + 2100$.

Si tttt = A806 en hexadécimal, tttt = 43014 en équivalent décimal,

la valeur correcte de la fréquence = $(43014/19)+2100=4363.89$ Hz.

Les coefficients a, b, c, d, et f_0 , fournis par le constructeur et calculés lors de l'étalonnage, sont utilisés dans la formule suivante:

$$\text{Température} = \frac{1}{\left[a + b \times \ln\left(\frac{f_0}{f}\right) + c \times \ln^2\left(\frac{f_0}{f}\right) + d \times \ln^3\left(\frac{f_0}{f}\right) \right]} - 273.15 \text{ } ^\circ\text{C}$$

La dérive du capteur de température ne dépend en aucune façon des conditions d'utilisation et ce dernier est insensible aux chocs. Elle est uniquement due au vieillissement du capteur et est généralement inférieure à 0.005 °C / an.

2.1.3.2. Calcul de la conductivité

La valeur corrigée de conductivité = $\text{sqrt}(\text{cccc} * 2100 + 6250000)$.

Si cccc = 03DA en hexadécimal, cccc = 986 en équivalent décimal,

la valeur de la fréquence = $(986 * 2100) + 6250000 = 2884.545$ Hz

Les coefficients a, b, c, d, et m, fournis par le constructeur et calculés lors de l'étalonnage, sont utilisés dans la formule suivante:

$$\text{Conductivité} = \frac{(af^m + bf^2 + c + dt)}{(10 \times (1 - 9.57 \times 10^{-8}) \times p)} \text{ Siemens/mètre}$$

Il est nécessaire de prendre quelques précautions élémentaires afin de maintenir la cellule de conductivité en bon état de fonctionnement. La mesure de conductivité dépend entre autres du facteur de forme de la cellule. Il est donc important de préserver cette dernière de tout dépôt biologique par application d'un antifouling puissant aux extrémités du tube en verre constituant la cellule.

Il est également important de préserver les électrodes de la cellule de toute contamination par des résidus huileux. Le constructeur conseille de re-platiniser les électrodes lors de chaque étalonnage.

L'instrument, lorsqu'il n'est pas en service, doit être rincé à l'eau douce, afin d'éviter tout dépôt de sel sur les électrodes. Il est recommandé après chaque voyage, de nettoyer la cellule avec un détergent (Triton X100 dilué à 1%) pendant plusieurs minutes et de la rincer proprement à l'eau douce.

Lors du stockage, la cellule doit être remplie avec de l'eau distillée et ses extrémités fermées avec un tube de tuyau "tygon" afin d'éviter toute contamination par des agents extérieurs.

Attention: Lors de l'envoi par avion de la sonde pour étalonnage, il ne faut pas mettre d'eau dans le capteur. Cette dernière risque de geler et de briser ainsi la cellule en verre rendant impossible tout post-étalonnage.

2.1.3.3. Calcul de la salinité

Le calcul de la salinité est obtenu à partir de l'échelle de salinité pratique établie en 1978 (IEEE J.O.E, 1980 et l'UNESCO, 1984). La salinité absolue est définie comme étant le rapport de masse de matière dissoute dans l'eau de mer sur la masse d'eau de mer. En pratique, cette quantité ne peut être mesurée directement et une salinité pratique est définie pour rendre compte des observations océanographiques. Cette salinité est une quantité sans dimension d'où le terme anglais utilisé habituellement: PSU (Practical Salinity Unit).

Le calcul de la salinité fait intervenir la conductivité électrique de l'eau de mer, la température et la pression:

$$R = \frac{C(S, T, P)}{C(35, 15, 0)}$$

R est le rapport de la conductivité in situ (S,T,P) sur la conductivité normale (35 ‰, 15°C, 0 dbar). C(35, 15, 0) = 4.2914 S/m.

$$R_T = \frac{R}{R_P r_T}$$

R_T est le rapport de conductivité de l'eau de mer, à la température T, à la conductivité de l'eau de mer ayant une salinité pratique de 35 ‰ à la même température.

R_p représente le rapport de la conductivité in situ sur la conductivité du même échantillon à la même température mais à pression P nulle. Dans le cas du thermosalinographe la mesure est réalisée à la pression atmosphérique, donc le terme $R_p = 1$.

$$R_p = 1 + \frac{P \times (A_1 + A_2 P + A_3 P^2)}{1 + B_1 T + B_2 T^2 + B_3 R + B_4 RT}$$

| | | | |
|---------|--------------------------|---------|-------------------------|
| $A_1 =$ | 2.070 E^{-5} | $B_1 =$ | 3.426 E^{-2} |
| $A_2 =$ | -6.370 E^{-10} | $B_2 =$ | 3.464 E^{-4} |
| $A_3 =$ | 3.989 E^{-15} | $B_3 =$ | 4.215 E^{-1} |
| | | $B_4 =$ | -3.107 E^{-3} |

r_T est le rapport de conductivité de l'eau de mer normale, ayant une salinité de $35 \text{ }^\circ/\text{oo}$, à la température T , sur sa conductivité à 15°C .

$$r_T = c_0 + c_1 T + c_2 T^2 + c_3 T^3 + c_4 T^4$$

| | |
|---------|-----------------------------|
| $c_0 =$ | 6.766097 E^{-15} |
| $c_1 =$ | 2.00564 E^{-5} |
| $c_2 =$ | $-1.104259 \text{ E}^{-10}$ |
| $c_3 =$ | -6.9698 E^{-15} |
| $c_4 =$ | 1.0031 E^{-5} |

La salinité pratique S est donnée par la formule suivante:

$$S = \sum_{j=0}^5 a_j R_T^{j/2} + \frac{(T-15)}{1+k(T-15)} \sum_{j=0}^5 b_j R_T^{j/2}$$

| | | | |
|---------|------------|---------|------------|
| $a_0 =$ | 0.0080 | $b_0 =$ | 0.0005 |
| $a_1 =$ | -0.1692 | $b_1 =$ | -0.0056 |
| $a_2 =$ | -25.3851 | $b_2 =$ | -0.0066 |
| $a_3 =$ | 14.0941 | $b_3 =$ | -0.0375 |
| $a_4 =$ | -7.0261 | $b_4 =$ | 0.0636 |
| $a_5 =$ | -2.7081 | $b_5 =$ | -0.01444 |

Un programme auxiliaire appelé **COMPUSAL**, livré avec la disquette programme, permet de calculer des salinités pratiques (PSU) à partir des données de conductivité, température et pression. Cet utilitaire est très utile lorsque l'on veut déterminer la dérive d'un appareil entre deux étalonnages. En effet, Sea Bird ne donne généralement que la dérive du capteur, en Siemens/mètre.

2.1.4. Interface

Le thermosalinographe Sea-Bird SBE-21 est fourni avec une liaison série bi-directionnelle (norme RS232) permettant d'acquérir les données et de configurer l'appareil.

2.2. Navigateurs par satellite

2.2.1. Magnavox MX-1107

Le récepteur satellite Magnavox MX-1107 équipe un navire océanographique de l'ORSTOM, l'Alis. Ce navigateur reçoit les données GPS ainsi que TRANSIT. Les données GPS sont prioritaires, mais en cas de perte de couverture, il permet d'obtenir les données du système TRANSIT. Ainsi, même pendant la couverture GPS, le Magnavox détecte les passages de satellite TRANSIT et en informe le programme. Ces informations sont stockées pour pouvoir être utilisées éventuellement en cas de perte de couverture GPS.

Le Magnavox est relié à l'ordinateur par l'intermédiaire d'une liaison série au format RS-232 (cf plus loin). Il envoie sur cette ligne une chaîne de caractères ASCII en hexadécimal. Le Magnavox envoie ces données en réponse à une demande venant de l'ordinateur, à l'exception des passages satellite TRANSIT qu'il envoie spontanément. A tout moment sont donc disponibles la dernière position GPS calculée ainsi que la dernière estime TRANSIT. Le Magnavox indique également l'état de la couverture GPS (point en trois dimensions, deux dimensions ou perte de la couverture). Cet indicateur permet au programme de basculer automatiquement en mode TRANSIT en cas de perte de couverture GPS.

2.2.2. Pronav GPS-100

Le récepteur Pronav GPS-100 est relié à l'ordinateur par une liaison série unidirectionnelle. Il transmet des données sous forme de sept chaînes de caractères ASCII, spontanément toutes les deux secondes. Le format de ces données est conforme à la norme **NMEA-0183**, qui est commune à de nombreux systèmes de navigation (récepteurs GPS, traceurs de route etc...). Cette norme précise à la fois un format de données et un niveau de signaux électroniques. Le respect de cette norme permet au logiciel de ne pas avoir à se préoccuper du type du récepteur GPS: tout appareil possédant une sortie NMEA-0183, quelle que soit sa marque ou son type sera utilisé de la même manière par le logiciel. Parmi toutes ces chaînes transmises, une seule nous intéresse: il s'agit de la trame **GPRMC**, qui nous indique l'heure GMT, la date, la position ainsi que l'état de la couverture GPS. En cas de perte de couverture, ce récepteur le signale mais n'entretient pas d'estime, il faut donc attendre le retour du GPS pour retrouver une position correcte. Dans ce cas, la seule solution est d'utiliser un autre moyen de navigation présent à bord du navire et d'entrer manuellement cette position (cf plus loin navigation manuelle).

2.2.3. Centrale NALNO

A bord du navire océanographique de l'IFREMER **Le Noroît**, les informations de navigation GPS sont fournies par la centrale de navigation **NALNO**. NALNO est un logiciel développé par l'IFREMER et fonctionnant sur un ordinateur HP9000 série 200. Il obtient les informations de navigation d'un récepteur GPS MLR, d'un récepteur LORAN-C et de deux récepteurs Magnavox TRANSIT et OMEGA. Il permet de piloter les récepteurs satellites TRANSIT en leur communiquant le cap et la vitesse du navire. Il fournit en fin de campagne une route corrigée et un tracé de carte. Ce système offre en outre une possibilité de connexion à partir un ordinateur IBM-PC par

l'intermédiaire de la liaison série prévue à l'origine pour l'ADCP (Acoustic Doppler Current Profileur). NALNO envoie sur cette ligne une trame GPGGA dont les informations ne sont pas suffisantes pour faire un suivi de route correct. En particulier, il manque la date. L'ORSTOM a donc demandé une modification des informations fournies par NALNO ainsi que la mise à disposition d'une autre liaison série, indépendante de l'ADCP, pour éviter tout conflit potentiel. Ceci n'a pu se faire qu'en utilisant la ligne dévolue au LORAN-C, inutilisée dans le pacifique. D'autre part, le format des données transmises par NALNO (trame GPGGA) est très proche du format NMEA-0183. Nous avons donc demandé que ce format devienne identique à la trame GPRMC. Cette modification est effective depuis octobre 1991, avec malheureusement quelques petites différences qui ont été corrigées en Février 1992.

Pour des raisons de compatibilité, le programme continue de gérer la trame GPGGA. Le format des trames GPRMC NALNO et GPGGA est donné en annexe ainsi que les paramètres de communication.

2.3. Navigation manuelle

Dans le cas où il n'est pas possible d'utiliser les moyens de navigation du navire, le programme permet de faire une navigation manuelle. A intervalles de temps réguliers, l'opérateur doit saisir la position, la date en jour julien et l'heure GMT de façon à avoir un minimum d'information de navigation.

3. CONNEXION DES APPAREILS

3.1. Liaison série: la norme RS232

Une liaison série est un des moyens les plus simples de transmettre de l'information entre deux appareils distants. En effet, ces appareils doivent être connectés par l'intermédiaire d'un câble très rudimentaire, devant comporter au minimum trois fils pour établir une communication bi-directionnelle: un fil de transmission, un fil de réception et un fil qui relie les deux masses logiques des appareils. Outre sa simplicité de mise en oeuvre, cette liaison présente l'intérêt d'être présente en standard sur les ordinateurs de type IBM-PC ou compatibles et sur un grand nombre d'appareils de mesures ou de récepteurs satellite.

Le principe de la transmission est lui aussi très simple: dans le système de représentation de données ASCII, chaque caractère est codé par une succession de huit bits dont la valeur est 0 ou 1. Ces bits sont envoyés les uns derrière les autres, par séquences de 5,6,7 ou 8 bits (*'longueur de mots'*). Cette méthode très simple nécessite cependant une phase de synchronisation entre les appareils. Une séquence de bits est donc entourée par un *'bit de start'* et un ou deux *'bits de stop'*, de façon à ce que le récepteur puisse connaître le début et la fin d'une séquence. Pour éviter les erreurs de transmission, un bit de contrôle, appelé *'bit de parité'*, optionnel est possible. La *'vitesse de transmission'* des données est paramétrable. Plus celle-ci est élevée, plus les risques d'erreur de transmission sont grands. Il faut donc déterminer la vitesse maximale acceptable en fonction de la qualité de la ligne reliant les appareils, la distance qui les sépare, les risques de rayonnements extérieurs risquant de perturber la transmission, ainsi que la vitesse de traitement de l'ordinateur de réception. Les vitesses fréquemment utilisées sont de l'ordre de 9600 bauds, soit 960 caractères par seconde, pour des distances allant jusqu'à plusieurs dizaines de mètres.

L'appareil récepteur n'a pas la possibilité de connaître les paramètres de communication utilisés par l'émetteur, il faut donc que ces paramètres soient fixés au départ, identiques sur les deux appareils.

Les caractères reçus par transmission série sont convertis dans un format utilisable par un microprocesseur, par l'intermédiaire d'un circuit intégré, l'**UART** (Universal Asynchronous Receiver Transmitter). Un mécanisme inverse est effectué lors de l'émission.

La norme **RS-232** définit un niveau de tension lors d'une transmission série. Les tensions de sortie d'une UART (0/+5V) ne sont en effet pas suffisantes pour franchir une distance supérieure à quelques mètres. Il faut se rappeler les correspondances entre ces deux logiques:

- un 0 TTL \Leftrightarrow +12 Volts RS 232.
- un 1 TTL \Leftrightarrow -12 Volts RS232.

Avec ces niveaux électriques, le signal peut parcourir plusieurs dizaines de mètres sans problèmes (15 mètres selon la norme mais il est pratiquement possible d'aller jusqu'à 100 mètres.

3.2. Multiplexeur de lignes RS232

3.2.1. Principe

Les ordinateurs que nous utilisons comportent en standard une ou deux liaisons série. La plupart des portables n'en ont qu'une. Cela veut dire qu'on ne peut brancher qu'un seul appareil sur l'ordinateur. Faire l'acquisition des données d'au moins deux appareils simultanément est alors impossible. La solution consiste à utiliser un *multiplexeur de lignes série*, qui permet de dupliquer une liaison série en deux, quatre voire huit ou seize lignes. Il existe des multiplexeurs très sophistiqués sur les ordinateurs centraux auxquels sont raccordés de nombreux périphériques. Nous n'avons pas besoin pour ce type d'application d'un tel matériel, relativement coûteux et généralement utilisable sous d'autres systèmes d'exploitation que MSDOS. Le laboratoire d'électronique du programme SURTROPAC a donc décidé de développer un multiplexeur, à la fois très simple de conception et bon marché.

3.2.2. Description technique et principe de fonctionnement

Pour schématiser, on peut comparer le fonctionnement du multiplexeur à celui d'un aiguillage. La voie principale étant connectée à l'ordinateur, le multiplexeur permet d'aiguiller le flux de données d'un appareil à la fois, parmi les quatre disponibles.

Le schéma du multiplexeur est proposé au chapitre 11.3.2. Les 5 paires d'inverseurs que l'on trouve en entrée et en sortie sur les lignes Tx (transmission) et Rx (réception), IC4 à IC6, servent d'adaptateur de niveau. Cette adaptation se fait dans le sens TTL/CMOS vers RS232 pour la ligne Rx et RS232 vers TTL/CMOS pour la ligne Tx. Ces circuits, des MAX232, peuvent générer les tensions +10 Volts et -10 Volts nécessaires au respect de la norme RS232, avec une seule alimentation de 5 Volts, moyennant l'utilisation de 4 condensateurs électrochimiques 10 μ F.

Le coeur du montage est un multiplexeur bi-directionnel 2 voies parmi quatre, IC1 (CD 4052). IC1 est commandé par les lignes de données D0 et D1 (port parallèle Centronics), au travers de l'octuple tampon trois états IC2 (74HC244). Les portes OU exclusif IC7, servent uniquement à éviter le fonctionnement du multiplexeur lorsque l'ordinateur est éteint. La ligne strobe commande IC2 et est active au niveau bas. Le décodeur d'adresse IC3 (74HC139), est utilisé pour la commande de la LED (D4 à D7) indiquant la voie sélectionnée (T3 à T6). Les deux transistors T1 et T2, commandent les LED D2 et D3. Ces dernières s'allument au rythme des caractères reçus ou envoyés.

3.2.3. Utilisation

L'utilisation dans un programme d'un tel multiplexeur est très facile. La sélection de la voie à activer se fait par l'intermédiaire du port parallèle Centronics. Une fois cette voie active, l'utilisation du port série est totalement identique avec ou sans le multiplexeur.

3.2.4. Avantages et inconvénients

Le principal inconvénient d'un multiplexeur comme le notre est qu'il n'autorise pas la communication simultanée avec l'ensemble de ses voies. A un instant donné, l'ordinateur ne peut réellement dialoguer qu'avec un seul appareil. Lorsque le débit est très important sur

l'ensemble des voies, le risque de perdre des données est grand. Dans notre application, cela n'est pas le cas et il suffit d'activer les voies les unes après les autres pour acquérir les données des différents appareils, sans risque de perte d'information.

Un autre inconvénient tient dans la façon de commander le multiplexeur. La sélection de la voie active se fait par l'intermédiaire du port parallèle de l'ordinateur, normalement réservé pour l'imprimante. Il est donc impossible d'utiliser une imprimante lorsque le multiplexeur est présent.

Ce multiplexeur a été développé dans le but d'offrir la possibilité de connecter plusieurs appareils sur les ordinateurs portables qui sont utilisés à bord des navires. Par ailleurs, la facilité de programmation de ce multiplexeur le rend très facile d'emploi dans un programme. Il fut donc facile d'intégrer ce nouvel élément dans le programme **THERMO**, prévu à l'origine pour fonctionner sans multiplexeur. Les inconvénients théoriques cités plus haut ne représentent pas de réelles restrictions d'emploi, puisque le port imprimante est inutilisé et que le débit des données n'entraîne pas de perte d'information.

D'un point de vue électronique, le multiplexeur convertit le signal en provenance d'un récepteur GPS conforme à la norme NMEA-0183 (0/+5V) en un signal conforme à la norme RS-232 (+12V/-12V). Il peut permettre de résoudre d'éventuels problèmes liés à l'altération des signaux lorsque la distance séparant les appareils devient supérieure à quelques mètres.

4. LES MODULES LOGICIELS

4.1. Les modes d'appel du programme

Le programme **THERMO** peut être lancé en spécifiant un certain nombre d'options sur la ligne de commande. Pour simplifier l'usage du programme et permettre une relance rapide lorsque la configuration n'a pas à être modifiée, l'appel de "thermo" sans paramètres nous permet d'accéder directement au module d'acquisition. "thermo -m" permet d'accéder au menu principal et donc de pouvoir re-configurer le programme. Le menu principal comprend quatre choix, décrits dans les paragraphes suivants.

4.2. Le module d'acquisition

Le module d'acquisition est entièrement automatique et ne nécessite aucune intervention de l'opérateur si on dispose d'un système de navigation par satellite. Dans le cas contraire, il faut saisir manuellement la position à intervalles réguliers (cf plus loin le mode le fonctionnement). Le fonctionnement de l'acquisition (période d'échantillonnage, fichier de sauvegarde etc...) est entièrement paramétrable (voir ci-dessous).

4.3. Le module de configuration du logiciel

Avant de lancer le module d'acquisition, le programme doit être configuré en fonction des appareils connectés et en fonction des besoins de la mission. La période d'échantillonnage, en particulier, est fixée par l'opérateur. Cette configuration s'effectue en remplissant un masque de saisie. Il est à noter que la configuration choisie est conservée d'un appel sur l'autre, ce qui facilite l'utilisation du programme: on ne doit modifier en général que très peu de paramètres (cf plus loin configuration).

4.4. Le module de configuration du thermosalinographe

Chaque thermosalinographe Sea-Bird SBE-21 doit être étalonné. L'ensemble des coefficients d'étalonnage est contenu dans un fichier ASCII. Il faut modifier ce fichier à chaque réétalonnage des capteurs et bien sûr à chaque changement de thermosalinographe (usage réservé à l'installateur du programme).

4.5. Le mode terminal

Le mode terminal permet de dialoguer directement avec les appareils connectés. C'est donc un bon outil de test des liaisons. La personne chargée de l'installation du programme au départ d'une mission pourra ainsi facilement détecter une erreur de branchement ou de configuration des liaisons. Cela permet en outre de configurer les appareils distants (cas du thermosalinographe).

L'usage 'normal' du programme nécessite juste de savoir modifier le nom du fichier de sauvegarde et de lancer le module d'acquisition, tous les autres paramètres ayant été au préalable définis, ils n'ont pas besoin d'être modifiés.

5. MISE EN OEUVRE DU LOGICIEL

5.1. Philosophie de développement

Conventions d'écriture

Pour clarifier la présentation des outils logiciels décrits dans la suite de ce manuel, nous utiliserons les conventions suivantes:

- Extrait de code en langage C POLICE Courier
- Nom des modules C, de programme ou de bibliothèque POLICE Bookman souligné
- Première apparition d'un mot clef POLICE **Bookman** gras
- Utilisation des touches du clavier POLICE *Bookman* italique

5.1.1. Une programmation multitâches

Un programme d'acquisition comme **THERMO** doit être capable d'acquérir simultanément des données en provenance de plusieurs appareils. En effet, on ne sait jamais à quel moment arriveront les données d'un récepteur **GPS100** ou **NALNO** par exemple, puisque ces appareils envoient des données suivant une fréquence qui leur est propre. Il en va de même lors d'un passage satellite **TRANSIT** pour un **Magnavox MX-1107**. Il faut donc que le programme soit en permanence à l'écoute de tous les appareils connectés. Dans un environnement multitâches, une solution naturelle consisterait à définir autant de processus qu'il y a d'appareils connectés. Chaque processus étant chargé de l'acquisition des données depuis un appareil et de leur transmission à un autre processus, chargé lui du traitement de ces données (affichage et sauvegarde sur disque par exemple). Malgré le fait que le système d'exploitation **DOS** n'offre pas cette possibilité, nous avons tout de même conçu le programme **THERMO** comme un programme multitâches. La solution consiste à écrire un programme fonctionnant, du moins en partie, sous **interruptions**.

Pour ne pas tomber dans l'écriture d'un système extrêmement complexe, nous n'avons pas tenté d'écrire un véritable environnement multitâches, comme peut l'être le système d'exploitation **UNIX** par exemple. L'essentiel étant d'être capable d'acquérir des données, nous nous sommes contentés de programmer par interruptions le module de réception de caractères sur un port série. Nous avons également besoin d'un module de gestion du temps fonctionnant de la façon la plus indépendante possible du programme. Nous avons utilisé là également une programmation par interruptions. Pour réaliser le reste du programme, à savoir un ensemble de tâches fonctionnant en parallèle, nous avons défini un **superviseur de tâches**, dont le rôle est de donner la main à chacune des tâches qui le demandent. Chaque tâche devant elle-même rendre la main au superviseur dans un délai raisonnable. Ce principe de base nous simplifie considérablement les choses, car la partie la plus délicate dans un vrai système multitâches est la mise au point d'un algorithme fiable permettant d'interrompre n'importe quelle action à n'importe quel moment pour en exécuter une autre, puis de reprendre l'action initiale là où elle s'était arrêtée. Cela veut dire également qu'on ne sait jamais où sera interrompue une tâche, puisque ceci dépend uniquement du nombre de tâches s'exécutant à un instant donné (une tâche seule ne serait jamais interrompue). Notre programme ne fonctionne pas du tout comme cela, car lorsqu'une tâche reçoit la main, elle doit la rendre au superviseur. De plus, contrairement à un vrai système multitâches, notre superviseur n'est pas capable d'exécuter

n'importe quel type de tâche. Toutes les tâches définies dans ce programme sont donc construites sur le même modèle.

Une autre notion très importante lorsqu'on écrit un système multitâches, est la communication des tâches entre elles. Par analogie avec le système UNIX, nous avons choisi d'utiliser la notion de **signaux**. Un signal est envoyé par une tâche à une autre tâche pour déclencher une action, lui indiquer la fin (correcte ou non) d'une action etc... Un signal ne permet pas de transférer de l'information à proprement parler, c'est simplement un code commun aux diverses tâches les utilisant. Nous avons défini un ensemble de signaux, chacun ayant une signification bien précise qui est connue de toutes les tâches.

5.1.2. Une programmation par objets

Nous avons écrit ce programme en langage C, qui n'est pas un langage à objets, mais qui permet tout de même d'en appliquer les principes les plus importants.

La programmation par objets est avant tout une manière d'écrire un programme de la façon la plus **modulaire** possible et de façon à ce que ses modules soient aussi **indépendants** les uns des autres que possible. Ces principes sont tout à fait adaptés à une programmation parallèle, puisque là aussi la notion d'indépendance est primordiale. La modularité et l'indépendance favorisent en outre l'évolution d'un programme, puisque l'écriture d'un module doit pouvoir se faire sans avoir à comprendre tout ce qui existe par ailleurs dans le programme. C'est d'autant plus important si l'on travaille en équipe.

Dans le vocabulaire de la programmation par objets, l'**objet** est la structure de données essentielle. Définir un objet, c'est donner deux choses:

- décrire la structure des données constituant l'objet (structure de données au sens habituel).
- définir l'ensemble des procédures permettant d'accéder à cet objet, appelées **méthodes**.

L'énorme intérêt de ce genre de représentation par rapport à l'approche algorithmique classique tient au fait qu'un objet est utilisable dans un programme sans que celui-ci ait besoin de connaître complètement sa structure. Un objet peut être considéré comme une **ressource** logicielle du programme, au même titre que le clavier ou l'écran sont des ressources matérielles. On n'a pas besoin de connaître exactement le mécanisme électronique et vidéo qui est mis en jeu pour écrire un mot à l'écran. Il faut considérer un objet exactement de la même façon. La seule connaissance réellement utile est l'ensemble des méthodes de manipulation de l'objet. La manière dont fonctionnent ces méthodes peut rester complètement inconnue, sans que cela nuise à l'utilisation de l'objet.

Dans un langage à objets, le programmeur voulant accéder à un objet est obligé d'utiliser les méthodes qui lui sont offertes avec cet objet. Il n'a pas directement accès aux données. Ceci est en outre un avantage pour le programmeur qui a défini cet objet: il peut en effet faire évoluer son implantation, sans que cela ait de conséquences sur le reste du programme, tant les méthodes garderont, **vues de l'extérieur** le même

comportement (c'est la moindre des choses!). Par contre un programme qui accéderait directement aux données de l'objet pourra être complètement bouleversé. En outre, le programmeur ayant défini un objet sait que celui-ci sera utilisé correctement, puisque accédé au travers de méthodes qu'il aura lui-même écrites. Ceci est un gage de fiabilité du programme.

Ces notions de programmation par objets ont été mises à profit dans le programme THERMO. Parmi les structures de données manipulées dans le programme, une seule a été définie réellement comme un objet, il s'agit de la représentation d'une **tâche**. Par contre, la façon d'écrire le programme, (modularité, indépendance) est directement issue de la programmation par objets.

5.1.3. Une programmation 'par couches successives'

La notion de **programmation par couches** est utilisée dans les systèmes de communications modernes (réseaux locaux, Ethernet TCP/IP etc...). Elle a été normalisée par l'ISO (International Standardisation Organisation) en sept niveaux, les sept couches OSI (Open System Interconnection).

Dans une telle représentation, une couche offre un **service** à la couche qui lui est immédiatement supérieure et ne peut utiliser que les ressources de la couche immédiatement inférieure. Chaque couche est responsable d'une petite partie du chemin que doit faire une donnée pour aller du niveau le plus bas (le matériel) jusqu'au niveau le plus élevé, appelé communément le niveau **application**. Une fois au niveau application, la donnée est mise en forme pour être manipulée.

Le programme THERMO est organisé en cinq couches successives, dont le fonctionnement est illustré par le cheminement à travers le programme, d'une donnée en provenance d'un appareil de mesure, arrivant sur une liaison RS232, jusqu'à son stockage sur disque. Nous n'avons pas implanté les sept couches OSI de manière stricte, mais nous avons repris les mêmes idées pour implanter cinq niveaux distincts.

- Acheminement d'un caractère depuis l'appareil distant: assuré par le matériel (UART pour une liaison série)
- Prise en compte d'un caractère sur l'ordinateur: c'est le plus bas niveau du logiciel. Ceci est assuré par le module **RS**. Il fonctionne sous interruptions pour permettre l'acquisition des caractères au fur et à mesure de leur arrivée, sans perturbation du programme. Le couple matériel/logiciel dépend du type de liaison utilisé. Il faudrait définir un autre couple pour gérer l'arrivée de données à partir d'une carte d'acquisition par exemple. Pour l'instant, le seul type de liaison que nous ayons eu à gérer pour développer le programme THERMO est la liaison série.
- Regroupement des caractères arrivés en trames: les caractères arrivés sont regroupés dans une **trame** formant un tout (ex: une position GPS). Ceci est assuré par le module **TRAME**. Le début et la fin d'une trame sont indiqués par la réception d'une séquence de caractères particuliers (très variable pour le début, en général <CR>/<LF> pour la fin). Ces trames sont mises en attente d'utilisation par le niveau supérieur et le module se remet en attente

jusqu'à ce qu'une nouvelle série de caractères arrive pour former une trame.

- **Décodage d'une trame:** Les données arrivant d'un appareil de mesure sont codées suivant un format qui lui est propre. Il faut décoder ces données de façon à les rendre utilisables par le reste du programme (affichage et stockage par exemple) et éliminer éventuellement les informations inutiles. Il existe dans le programme un module de décodage par appareil. Une fois décodées, les données sont mises en attente d'utilisation. Le module se remet en attente de l'arrivée d'une nouvelle trame complète.
- **Affichage et stockage des données:** c'est le niveau application cité plus haut. Le module attend les données des différents appareils, les regroupe en un seul enregistrement qui est stocké sur disque.

5.2. Les bibliothèques logicielles

5.2.1. Une gestion de ports série sous interruptions

5.2.1.1. Principes généraux

Le but de ce module est de gérer la communication à travers un port série d'un IBM-PC ou compatible. Le système DOS est capable de gérer deux ports série seulement (**COM1/COM2**), bien que quatre soient prévus par le matériel (**COM1..COM4**). La raison est que deux niveaux d'interruptions (**Interrupt ReQuest** ou **IRQ**) seulement sont associés aux ports série (il en faudrait quatre): ce sont **IRQ4** pour **COM1** et **IRQ3** pour **COM2**. Il est possible de résoudre ce problème, mais nous n'en parlerons pas ici, car aucune des machines sur lesquelles nous voulons installer le programme n'a plus de deux ports série.

5.2.1.2. Les interruptions matérielles

Le principe général d'une interruption matérielle est simple: lorsqu'un événement matériel se produit, le système peut, s'il a été programmé pour cela, déclencher une interruption (appelée alors interruption matérielle). Cette interruption n'est pas reçue directement par le microprocesseur, mais par un circuit intégré spécialisé dans la gestion des interruptions matérielles, le **PIC** ou **Programmable Interrupt Controller** dans un PC. Le rôle principal d'un tel circuit est de gérer les conflits possibles entre les différentes interruptions susceptibles de se produire au même instant. Pour ce faire, on définit une hiérarchie dans les interruptions, ce sont les **IRQ** dont il est question plus haut. L'interruption de niveau 0 est la plus prioritaire. Sur un IBM-PC ou compatible, huit niveaux d'interruption sont définis (de 0 à 7), seize sur un AT (0 à 15). Cette hiérarchie est définie par le matériel, il est donc impossible de la modifier. Lorsque le PIC a résolu les éventuels conflits d'interruption, il envoie un signal d'interruption au microprocesseur, qui n'avait pas encore été concerné. Ceci a pour conséquence l'arrêt immédiat de la séquence d'instructions que le microprocesseur était en train d'exécuter (le programme en cours), la sauvegarde du contexte pour pouvoir reprendre l'exécution et le déroutement vers le gestionnaire logiciel de l'interruption, c'est à dire le branchement à une adresse mémoire prédéfinie. A chaque niveau d'interruption est associée une adresse mémoire qu'il n'est pas possible de modifier. Les traitements standards associés à ces interruptions sont vides. Ecrire

un programme de gestion d'une interruption, c'est installer à l'adresse mémoire associée à cette interruption son propre code. Ce code sera alors exécuté, suivant le processus décrit plus haut, à chaque fois qu'une interruption d'un niveau d'IRQ donné se produira. Ecrire un gestionnaire pour le port série COM1, c'est donc installer du code à l'adresse correspondant à IRQ4. Lorsque le code d'interruption est terminé, le microprocesseur reprend le cours normal du programme jusqu'à la prochaine interruption.

Remarque: Un facteur déterminant pour la conception d'un gestionnaire d'interruption est le temps d'exécution. Il faut faire très attention à la dégradation des performances de la machine.

5.2.1.3. Cas des ports série

Le port série COM1 est associé au niveau d'interruption 4 (IRQ4). L'adresse mémoire est donc $4 \times 4 = 16$. Cette adresse doit contenir une instruction de saut vers la partie de code que l'on veut installer. Pour le port série COM2, associé au IRQ3, il faut installer notre code à l'adresse $4 \times 3 = 12$. Dans le cas des ports série, quatre événements sont susceptibles de déclencher une interruption: la réception d'un caractère, l'envoi d'un caractère, la modification de l'état de la ligne et du modem.

La première chose que doit faire un gestionnaire d'interruption est d'identifier le type de l'interruption parmi ces quatre possibilités. Une modification de l'état de la ligne indique une erreur de transmission, l'état du modem permet de gérer en particulier le contrôle de flux matériel entre les appareils. Le traitement typique à réception d'un caractère est de le stocker dans une zone mémoire accessible au programme, qui pourra ainsi le lire le moment venu. L'interruption correspondant à l'émission d'un caractère se produit quand le matériel a effectivement transmis le caractère. Il indique ainsi qu'il est prêt à en émettre un autre. On peut dans ce cas, soit envoyer le caractère suivant s'il existe (c'est à dire le premier en attente dans un tampon d'émission), soit simplement indiquer au programme que le port série est disponible (positionnement d'un drapeau en mémoire).

5.2.1.4. Gestion du multiplexeur

D'un point de vue logiciel, la présence d'un multiplexeur ne nécessite que peu de modifications dans la gestion d'un port série. La gestion proprement dite du port reste identique, il a suffi de revoir la phase d'initialisation de l'UART. Initialiser une UART, c'est lui envoyer un code définissant les paramètres de communication à utiliser et c'est installer son gestionnaire d'interruption. Cette initialisation reste valide jusqu'à la fin du programme ou la prochaine initialisation. Lorsque le multiplexeur n'est pas présent, chaque appareil est connecté sur son propre port série (COM1 ou COM2), géré de façon indépendante par deux gestionnaires d'interruptions distincts. Par contre, si ces appareils sont connectés sur un multiplexeur, du point de vue de l'ordinateur, il n'y a qu'un seul port série (COM1 en général). Il faut donc distinguer par logiciel les différents ports. La tâche est rendue très simple par le fait qu'à un instant donné, une seule des voies du multiplexeur est active (voir plus haut description du multiplexeur). L'activation d'une voie se fait par logiciel, elle consiste à envoyer au multiplexeur le numéro de la

voie à activer via le port parallèle, et à programmer l'UART de l'ordinateur (paramètres de communication, installation du code d'interruption). Chacun des appareils ayant en effet des paramètres de communication qui lui sont propres, il faut reprogrammer l'UART du PC avec ces paramètres, ceci à chaque bascule.

5.2.1.5. Les fonctions d'interface

Un port série est décrit par la structure de données suivante :

```
struct {
    RS_IDENT id;
    RS_VOIE   voie;
    PARAM     config;
    RING      inbuff,
             outbuff;
    PHYS      *phys;
} SERIE;
```

où

RS_IDENT = COM1 ou COM2.

RS_VOIE = VOIE1 .. VOIE4 ou NOMUX.

PARAM est une structure contenant les paramètres physiques du port (vitesse, parité etc..).

RING est un tampon circulaire.

PHYS est une structure contenant les adresses physiques du port et l'adresse du code d'interruption.

Les vitesses de transmission acceptées par le programme sont:

110,300,600,1200,2400,4800,9600,19200,38400,57600 et 115200 bauds.

La transmission peut être SANS PARITE ou de parité PAIRE ou IMPAIRE (type TPARITE).

Le nombre de bits de données peut être 5,6,7 ou 8.

Le nombre de bits de stop est 1 ou 2.

L'échange de données entre le programme principal et le module de gestion des ports série (RS) se fait par l'intermédiaire des deux tampons circulaires inbuff et outbuff. Quand RS reçoit un caractère depuis le port série, il le stocke dans inbuff et quand le programme principal veut envoyer un caractère sur le port, il le stocke dans outbuff.

Voici les fonctions permettant de mettre en oeuvre cet algorithme:

```
- SERIE *rsopen( RS_IDENT id_com,RS_VOIE no_voie,
                unsigned vitesse,TPARITE parite,
                unsigned char data,unsigned char stop,
                procedure (*mcr_hdl)() );
```

rsopen() permet d'initialiser un port série. C'est donc la première fonction à appeler. Les paramètres indiquent le port physique de connexion (COM1 ou COM2), le numéro de la voie du multiplexeur (VOIE1 à VOIE4 ou NOMUX si pas de multiplexeur), les paramètres de communication ainsi qu'une procédure à appeler en cas de

modification de l'état du modem. Ceci permet de gérer correctement le contrôle de flux matériel. Il faut passer NULL si aucun contrôle de flux n'est nécessaire. `rsopen()` retourne un pointeur de type `SERIE *`, qui est utilisé par toutes les fonctions suivantes. `rsopen()` initialise l'UART avec les paramètres de communication, installe le code de gestion de l'interruption du port série concerné et configure le multiplexeur pour activer la voie (si différente de `NOMUX`). Pour une utilisation avec le multiplexeur, `rsopen()` doit être appelée pour chaque voie, puisqu'elles sont considérées comme des ports série complètement indépendants.

- procédure `rsclose(SERIE *rsp);`

`rsclose()` permet de terminer proprement une session de travail avec un port série. Cette procédure doit être appelée en fin de programme pour tous les ports ouverts. Le paramètre à passer est le pointeur retourné par `rsopen()`.

- procédure `rputc(SERIE *rsp, unsigned char c);`

`rputc()` envoie un caractère sur le port série pointé par `rsp`. Si le port série est libre, le caractère est immédiatement envoyé, sinon il est stocké dans le tampon d'émission `outbuff` pour un envoi sous interruption.

- procédure `rputs(SERIE *rsp, unsigned char *s);`

`rputs()` fonctionne de la même façon que `rputc()` mais permet l'envoi d'une chaîne de caractères.

- int `rgetc(SERIE *rsp);`

`rgetc()` lit le premier caractère se trouvant dans le tampon de réception `inbuff`. Retourne -1 si aucun caractère n'est présent dans le tampon.

- procédure `pr_serie(SERIE *rsp);`

`pr_serie()` est une procédure utilisée en phase de mise au point (debug en Anglais) qui affiche les paramètres de communication d'un port série. Elle permet de savoir si l'appel de `rsopen()` était correct.

- procédure `rsoff();`

`rsoff()` permet d'interrompre la transmission de tous les ports série. Ceci permet de rendre l'ordinateur complètement 'sourd' au monde extérieur. Cette fonction est utilisée dans le programme `THERMO` quand une saisie manuelle de la position du navire est demandée. Le programme interrompt ses acquisitions de données le temps de la saisie.

- procédure `rson();`

`rson()` permet de reprendre l'écoute des ports série actifs, mais stoppés par `rsoff()`.

Les fonctions suivantes sont sans effet si on n'utilise pas de multiplexeur, mais peuvent être appelées sans problème de façon à avoir un programme identique dans les deux cas.

- procedure **rsactive**(SERIE *rsp);

rsactive() programme le multiplexeur pour rendre active la voie gérée par **rsp**. Cette procédure est essentielle pour utiliser le multiplexeur. Il faut appeler cette fonction lorsque l'on veut dialoguer avec l'appareil connecté sur le port série pointé par **rsp**, car par défaut aucune des voies du multiplexeur n'est active.

- procedure **rsdefault**(SERIE *rsp);

Indique au programme quelle voie du multiplexeur doit être activée par défaut. Ceci est utile lorsqu'on veut que le programme soit prioritairement en acquisition d'un appareil (la navigation pour le programme THERMO). Dès l'acquisition d'un autre appareil terminée, le programme se remet à l'écoute de cette voie.

- procedure **rsdesactive**(SERIE *rsp);

rsdesactive() désactive le multiplexeur. Si une voie a été déclarée par **rsdefault()**, cette voie est réactivée, sinon plus aucune voie n'est active après l'appel de **rsdesactive()**.

- booleen **rslock**(SERIE *rsp);

rslock() pose un verrou sur une voie du multiplexeur. Ceci interdit la pose d'un autre verrou et permet donc de s'approprier le multiplexeur. Cette fonction retourne **TRUE** si le verrou peut être posé et **FALSE** sinon. Ceci permet de garder de façon sûre la liaison avec un appareil. Si on n'utilise pas de multiplexeur, **rslock()** retourne systématiquement **TRUE**.

- procedure **rsunlock**(SERIE *rsp);

Libère le verrou posé par **rslock()**. Le couple **rslock()/rsunlock()** doit encadrer une session de dialogue avec un appareil. Tant que **rslock()** retourne **FALSE**, cela veut dire qu'on ne peut pas ouvrir la communication avec l'appareil. Une fois le dialogue terminé, il faut libérer la ligne par **rsunlock()**.

5.2.1.6. Le niveau TRAME

Le module **RS** permet d'acquérir des données sous forme de caractères. Dans la plupart des cas, les données sont regroupées en trames, formant un tout. On ne peut traiter ces données que lorsque l'on a reçu une trame complète. Ceci est le cas pour tous les appareils que nous avons eu à traiter jusqu'ici. Nous avons donc écrit un module de gestion de trames, construit au dessus de **RS**. Une trame est une zone tampon en mémoire, dans laquelle s'accumulent les données et qui est mise à disposition de la tâche d'acquisition lorsque toutes les données sont reçues. De même qu'il existe une structure de données de type **SERIE** pour chaque port série géré dans le programme, il existe une structure de données **TRAME** par port série:


```

struct {
    PAQUET *pret,
           *traitmt;
    char   etat;
    ERROR  erreur;
} TRAME;

```

PAQUET définit une zone mémoire de 80 caractères.

Voici les primitives d'accès à une trame:

```
- procedure init_trame( TRAME *trame );
```

Toutes les trames définies dans le programme viennent 'piocher' dans un même pool de PAQUET. `init_trame()` permet d'initialiser ce pool si ce n'est pas déjà fait et d'initialiser la trame.

```
- procedure lib_pool();
```

Libère la mémoire qui a été allouée par `init_trame()`.

```
- booleen new_trame( TRAME *trame );
```

S'alloue un paquet dans le pool des paquets disponibles. Retourne TRUE si elle trouve un paquet libre et FALSE sinon. Cette fonction est appelée à la réception du premier caractère de la trame (qui est en général un caractère particulier).

```
- procedure send_appli( TRAME *trame );
```

A la réception du dernier caractère d'une trame (<LF> en général), le contenu de la trame est mis à disposition de la tâche d'acquisition par `send_appli()`.

```
- procedure lib_trame( TRAME *trame );
```

Une fois le contenu d'une trame envoyé à la tâche d'acquisition, il faut se remettre en attente de nouvelles données en appelant `lib_trame()`.

```
- PAQUET *paquet_pret( TRAME *trame );
```

Retourne un pointeur non NULL si une trame complète a été reçue, c'est à dire si `send_appli()` a été appelée pour cette trame. Le paquet pointé contient les données reçues (format chaîne de caractères terminée par '\0').

```
- procedure lib_paquet( PAQUET *paquet );
```

Remet un paquet dans le pool des paquets disponibles.

```
- procedure reset_paquet( PAQUET *paquet );
```

Initialise un paquet vide. A appeler en début de trame, juste après `new_trame()`.

```
- procedure put_paquet( char c, PAQUET *paquet );
```

Ajoute un caractère en fin de paquet.

5.2.2. Une gestion de compteurs sous interruptions

5.2.2.1. Les besoins

Il est indispensable qu'un programme comme THERMO ait la connaissance du temps écoulé. En effet, la base d'un tel programme est de déclencher un certain nombre de traitements à des intervalles réguliers, aussi précis que possible. De plus, dans tout programme de communication, il est nécessaire que le programme puisse se rendre compte d'un problème matériel au niveau de la liaison. Il n'existe malheureusement pas de moyen de connaître l'état d'une liaison série (ou parallèle d'ailleurs). Il n'y a aucun moyen non plus de savoir si un caractère est effectivement bien arrivé sur la machine distante. C'est pour cela que des protocoles logiciels ont été écrits, prévoyant, entre autres, l'envoi d'accusés de réception et/ou une déconnexion en cas de non réponse au bout d'un temps fixé par le protocole. Il est en effet important que le programme ne reste pas 'planté' en cas d'interruption de la liaison. L'élément indispensable pour résoudre ce genre de problème est la connaissance du temps écoulé.

5.2.2.2. Principe

Il existe dans les compatibles IBM-PC un circuit intégré (référence: timer 8253) dont un des modes de fonctionnement permet de générer une interruption matérielle à une fréquence *proche* de 18.2Hz. Il est très facile alors d'écrire un code d'interruption, dont le rôle sera d'incrémenter un compteur à chaque interruption. On déduit ainsi rapidement de la valeur contenue dans ce compteur, la durée écoulée depuis son lancement. La fréquence standard de 18.2Hz nous offre une précision de 55ms environ. Cette précision est largement suffisante (voire trop importante) pour notre application. Il est tout de même possible de modifier la fréquence de génération de l'interruption si cela s'avère nécessaire. Elle est déduite de la fréquence d'oscillation d'un quartz de 1,1931816 MHz, divisée par un mot de 16 bits (de 0 à 65535). La fréquence minimale est $18.2 = 1,1931816/65535$, la fréquence maximale est la fréquence du quartz lui-même. Il est par contre impossible d'abaisser la fréquence en deçà de 18.2Hz, ce qui serait pourtant utile car dans le programme THERMO par exemple, la précision souhaitée étant de l'ordre de la seconde. Il est bien entendu que plus la fréquence d'interruption est élevée, plus le temps d'exécution du programme devient critique. Une façon d'optimiser la charge de travail du microprocesseur serait de ne pas le surcharger en générant des interruptions inutiles. Ceci n'étant malheureusement pas possible sur un IBM-PC, nous avons dû optimiser l'écriture du code d'interruption au maximum.

5.2.2.3. Les fonctions d'interface

Les fonctions ci-dessous se trouvent dans le module DELAI.

```
- procedure init_delais( unsigned char nb_delais,  
                        float frequence, float precision);
```

init_delais() est la première procédure à appeler pour utiliser la bibliothèque de gestion des timers. nb_delais contient le nombre maximal de timers à gérer dans le programme. Minimiser ce nombre

permet d'optimiser le temps d'exécution du code d'interruption. `frequence` est la fréquence à laquelle on doit programmer l'interruption matérielle pour obtenir la précision voulue par le programme. Si `frequence=0`, l'interruption sera générée à une fréquence de 18.2Hz. Dans le cas où cette fréquence serait encore trop élevée, on peut indiquer (en Hz) la précision voulue pour ces timers (1 donne une précision de 1 seconde, 18 une précision d'environ 55 ms).

```
- procedure term_delais();
```

Procédure permettant de terminer proprement le module de gestion des timers. Il est indispensable d'exécuter cette fonction pour laisser le système dans un état correct.

```
- procedure arme_delai( unsigned char no_delai,  
                      unsigned long nb_tops );
```

Il existe deux types de compteurs: les comptes à rebours, initialisés à une valeur de `nb_tops > 0`, qui est décrémente jusqu'à 0, et les compteurs simples, dont la valeur de départ est 0 et qui s'incrémentent tant que le compteur tourne. Les compteurs sont référencés par leur numéro (à partir de 0). Le compteur est en route dès l'appel de `arme_delai()`. La valeur `nb_tops` est un nombre d'interruptions. La durée réelle correspondant à cette valeur dépend de la fréquence choisie dans `init_delai()`. Le programme peut néanmoins calculer facilement la valeur à passer à `arme_delai()` pour armer le compteur pour une durée déterminée. Une fois que `init_delais()` a été appelée, une variable globale, `float FREQ_HORLOGE`, est construite. Elle contient le nombre d'interruptions générées en une seconde.

Exemple: pour armer un compteur d'une durée de 3.5 secondes, il faut passer une valeur de `nb_tops=3.5 x FREQ_HORLOGE`.

```
- procedure desarme_delai( unsigned char no_delai );
```

Procédure permettant de stopper un compteur.

```
- unsigned long valeur_compteur( unsigned char no_delai );
```

Cette fonction retourne la valeur du compteur `no_delai`. Pour convertir cette valeur en durée (secondes), il faut appliquer la formule inverse à celle présentée pour `arme_delai()`.

Exemple: une valeur de 1000 donnera une durée de `1000/FREQ_HORLOGE` secondes.

```
- booleen a_rebours( unsigned char no_delai );
```

Cette fonction retourne TRUE si le compteur considéré est un compte à rebours.

```
- TETAT etat_delai( unsigned char no_delai );
```

Le type TETAT est le type énuméré suivant: {ARME, DESARME, CLAQUE}. Après `init_delais()`, tous les compteurs sont dans l'état DESARME. Une fois armé, un compteur passe dans l'état ARME. S'il a été armé avec une valeur nulle, il restera armé tant qu'on ne le stoppera pas; s'il a été armé

avec une valeur > 0 (compte à rebours), le compteur restera armé jusqu'à ce que sa valeur devienne nulle. Il passera alors dans l'état `CLIQUE`. `etat_delai()` permet de connaître à tout moment l'état d'un compteur. Cet état n'est pas modifié.

```
- boolean delai_claque( unsigned char no_delai );
```

Cette fonction ne s'applique que pour les comptes à rebours. Elle retourne `TRUE` si l'état du compteur est `CLIQUE` et `FALSE` sinon. Lorsque `delai_claque()` retourne `TRUE`, l'état du délai passe de `CLIQUE` à `DESARME`, si bien qu'un deuxième appel de `delai_claque()` sur ce même compteur retournera `FALSE`.

5.2.3. L'horloge temps réel des IBM-AT

5.2.3.1. Description

Les micro-ordinateurs à base de microprocesseur 80286, 80386 se différencient des PC-XT (8086,8088) par de nombreuses améliorations, dont la présence en standard d'une horloge entretenue par une batterie. Elle permet naturellement d'entretenir la date et l'heure, en cas d'arrêt de la machine, d'où son nom de *horloge temps réel*. Cette horloge fait partie d'un circuit intégré, appelé RT/CMOS (Real-Time Clock/Complementary Metal Oxide Semiconductor). Ce circuit comporte en outre une zone mémoire de 50 octets, où il conserve la configuration matérielle de la machine (le setup). L'accès à ces informations s'effectue aux adresses 0x70 et 0x71. On peut ainsi à tout moment aller lire la configuration de sa machine, connaître son type et consulter l'heure et la date. Ces deux informations sont les plus intéressantes pour nous ici, car elles vont nous permettre de régler un problème ennuyeux qui se pose sur les IBM-PC.

5.2.3.2. Le rôle dans le programme

Nous avons décrit dans le module précédent la manière de gérer des compteurs indépendants de la vitesse d'exécution de la machine. Le principe que nous décrivons là est utilisé par le système lui-même pour entretenir l'heure et la date (commandes DOS `time` et `date`). Nous pouvons très vite constater que l'heure donnée par `time` dérive par rapport à une horloge de référence (heure satellite par exemple). Il faut donc recalibrer périodiquement la valeur si on ne veut pas que le décalage devienne très importante (en moyenne 1 seconde / heure). Dans un contexte d'acquisition de données, il n'est pas possible d'interrompre le programme pour recalibrer l'horloge. Il faut donc que le programme lui-même puisse faire ce travail. Autant dire tout de suite que cela n'est pas possible sur un PC-XT. L'horloge temps réel n'existe pas en standard et quand elle existe, son adresse d'implantation diffère d'une machine à l'autre. Il n'est pas possible de faire un programme fiable dans ces conditions. Par contre, sur un AT, la présence du circuit RT/CMOS permet de résoudre ce problème. Nous avons en effet remarqué que l'heure de l'horloge entretenue dérive peu (1 seconde / jour environ). Cette précision est amplement suffisante. Il suffit de recalibrer cette horloge au début de la mission et à chaque escale pour éviter toute dérive importante. Nous utilisons donc cette horloge temps réel comme référence de temps et périodiquement, nous mettons à jour l'heure et la date système à partir de cette référence.

5.2.3.3. Les fonctions d'interface

Toutes les fonctions décrites ici utilisent la structure de données suivante:

```
struct {
    unsigned        année;
    unsigned char   mois,
                  jour,
                  heure,
                  minute,
                  seconde;
} CMOS_D_T;

- CMOS_D_T *get_cmos_date_time();
```

Construit et retourne un pointeur sur une structure CMOS_D_T, contenant l'heure et la date provenant de la mémoire CMOS. La valeur retournée pointe sur une zone statique, dont le contenu est écrasé à chaque appel de la fonction.

```
- CMOS_D_T *get_pc_date_time();
```

La structure de données construite est la même que pour la fonction précédente, mais elle contient les informations système (identique aux commandes date et time). Ici aussi, la valeur retournée pointe sur une zone statique.

```
- procedure set_pc_date_time( CMOS_D_T *cmos_d_t );
```

Met à jour l'heure et la date système à partir des informations contenues dans la structure pointée par cmos_d_t (typiquement issue de get_cmos_date_time()).

Il existe également dans la mémoire CMOS une fonction permettant de générer une interruption périodique. La fréquence de cette interruption est de 1024 Hz, non modifiable. Nous avons donc à notre disposition une horloge extrêmement précise (environ 0.9765ms). Cette horloge ne peut par contre gérer qu'un seul compteur. Nous pouvons alors écrire un ensemble de procédures, semblables à celles du module DELAI, pour utiliser cette horloge. Nous ne décrivons pas ce module plus précisément, car nous ne l'utilisons pas dans le programme THERMO. Le lecteur pourra se reporter au module CMOS pour de plus amples détails.

5.3. Un ensemble de tâches communicantes

Nous avons décrit plus haut dans ce document les axes de développement du logiciel. Nous allons voir ici plus précisément l'organisation des modules, leurs rôles respectifs. Le programme peut être découpé en trois niveaux.

5.3.1. L'organisation du programme

Les modules de bas niveau que nous venons de présenter (TRAME, RS, DELAI, CMOS) sont des modules qui peuvent être utilisés dans divers programmes. Ils constituent le niveau 1.

Nous avons bâti au dessus de cet ensemble une bibliothèque de fonctions (SAT), destinées à un programme d'acquisition de données en temps réel. On trouve dans cette bibliothèque de niveau 2, les fonctions de base nécessaires à la définition des tâches dont il est question plus haut (modules TACHES et SCHED). On y trouve également les modules d'acquisition des données de navigation (modules MVX, NMEA183 et NALNO), ainsi que leurs modules de simulation (SIMNALNO, SIMULMVX et SIMUNMEA). Cette bibliothèque peut être utilisée par tous les programmes d'acquisition. Pour l'instant, deux logiciels sont basés sur cette bibliothèque: THERMO bien sûr et GEOACQ, logiciel destiné à l'UR de géophysique, qui fait l'acquisition de la magnétométrie sous-marine. Ce logiciel est utilisé lors de campagnes en mer, il utilise de ce fait les mêmes instruments de navigation que THERMO. Un troisième est à l'étude, dont le but serait l'acquisition des données météo, couplées aux données de salinité de surface.

Le troisième niveau est constitué des modules qui varient d'un logiciel d'acquisition à un autre: instruments de mesure spécifiques (thermosalinographe d'un côté, magnétomètre de l'autre), formats des fichiers de données, présentation de l'information à l'écran, possibilités d'interaction par l'opérateur etc...

5.3.2. Les tâches

La tâche est le concept de base du programme. On distingue plusieurs types de tâches. Les tâches d'acquisition, dédiées à un instrument de navigation dont le rôle est d'acquérir via le niveau 1 les données de l'appareil distant, de gérer les erreurs de communication, de décoder ces données et de les mettre à la disposition des autres tâches. Ces tâches sont décrites dans les modules MVX, NMEA183, NALNO, THERMO. D'autre part, on trouve dans le programme une tâche chargée d'afficher à l'écran les données au fur et à mesure de leur arrivée (module PRESENTE), une tâche qui déclenche les acquisitions et qui s'occupe de la sauvegarde des données (module SVGRDE), une tâche d'affichage de l'état des appareils distants (module STATUT), une tâche d'affichage du compteur indiquant le temps qui reste jusqu'à la prochaine acquisition (module TIMERS), une tâche de scrutation du clavier (module CLAVIER) et enfin la tâche **superviseur** (module SCHED), dont le rôle est de donner la main aux différentes tâches.

On peut ajouter une autre liste de tâches, ce sont les tâches de simulation, chargées de remplacer un appareil. Il en existe pour tous les appareils gérés par le programme. Nous n'en parlerons pas ici.

5.3.3. Les signaux

Toutes ces tâches n'auraient que de peu d'intérêt si elles ne pouvaient pas échanger d'information entre elles. Dans ce programme, l'échange d'information se fait par l'intermédiaire de signaux. Dans le paragraphe précédent, nous indiquons qu'une tâche d'acquisition met les données qu'elle a reçu à la disposition des autres tâches. En fait, elle indique par l'envoi d'un signal à certaines tâches, qu'elle a des données prêtes à être traitées. Un signal est envoyé à une seule tâche à la fois (répéter l'envoi pour toucher plusieurs tâches). Si une tâche n'a pas prévu de traiter le signal reçu, aucun traitement n'est effectué, aucune erreur n'est générée. Il faut donc définir la liste des signaux pertinents pour une tâche au moment de son écriture et savoir dans quelles conditions elle sera susceptible de recevoir un signal. Voici la liste des

signaux que nous utilisons (type énuméré SGN) avec leur signification (code respecté par toutes les tâches):

- **SIGKILL**: permet de tuer une tâche. C'est ainsi que l'on termine le programme: lorsque l'on appuie sur *Ctrl-F1*, la tâche clavier envoie ce signal à la tâche sched, ce qui a pour conséquence de terminer l'acquisition.
- **SIGSTART**: envoyé en début de programme.
- **SIGTERM**: envoyé en fin de programme. Ces deux signaux ne sont envoyés qu'à une seule tâche (svgrde), par la tâche sched juste avant et juste après l'acquisition. Ceci est utilisé dans le programme THERMO pour écrire dans le fichier de données une ligne indiquant le début et une indiquant la fin de l'acquisition. On peut donc ainsi facilement extraire les différents legs d'une mission, contenus dans un même fichier.
- **SIGBUSY**: signal envoyé par une tâche d'acquisition dans le cas où une demande d'acquisition lui arrive alors que la précédente n'a pas encore pu être honorée. Cela ne devrait pas arriver dans un déroulement normal.
- **SIGASK**: signal envoyé à une tâche d'acquisition pour déclencher une mesure.
- **SIGASW**: signal envoyé par une tâche liée à un appareil de mesure (pas de navigation), pour indiquer qu'une valeur est disponible. Réponse normale après réception d'un **SIGASK**.
- **SIGGPS**: signal envoyé par une tâche d'acquisition GPS, quand la position GPS est disponible.
- **SIGSAT**: signal envoyé par une tâche de navigation TRANSIT, au passage d'un satellite.
- **SIGFIX**: fait suite au précédent, lorsque les données en provenance du satellite sont disponibles.
- **SIGEST**: signal envoyé par une tâche de navigation TRANSIT, quand la position d'estime est disponible.
- **SIGERR**: Envoyé en cas d'erreur lors de l'acquisition. Valable pour toutes les tâches d'acquisition.

5.3.4. Fonctionnement d'une tâche

Le comportement d'une tâche varie selon le rôle qu'elle doit jouer, mais son mode de fonctionnement doit être le même pour simplifier l'écriture du superviseur.

Au lancement de l'acquisition, le superviseur initialise toutes les tâches qui ont été préalablement définies (méthode `init()`) (voir plus loin `define_tache()`). Le programme a la possibilité de donner la main à une tâche particulière (une seule), dès l'initialisation des tâches effectuée et avant le lancement de l'acquisition proprement dite. Dans le programme THERMO, c'est la tâche de stockage des données qui prend la main ici pour indiquer dans le fichier de données le début d'une série d'acquisition. Symétriquement, une tâche prendra la main dès

l'acquisition terminée. Là encore, il s'agit de la tâche de stockage qui indique dans le fichier de données la fin de l'acquisition. Ensuite, les tâches se terminent 'proprement', par l'exécution d'une procédure de terminaison (méthode `term()`) qui leur est propre (correspondant à la notion de **destructeur** en termes de programmation par objets).

En phase d'acquisition, le superviseur exécute à tour de rôle pour chaque tâche la méthode `polling()`. Ainsi, périodiquement, chaque tâche pour laquelle cette méthode existe prend la main et exécute cette procédure. La tâche perd la main à la fin de la procédure. Une tâche dont la méthode `polling()` est nulle ne peut pas prendre la main périodiquement. Dans le programme **THERMO**, il existe des tâches construites comme cela. La tâche `statut` par exemple n'a pas besoin de prendre la main régulièrement, mais seulement quand une autre tâche lui indique un changement d'état. Par contre, une tâche d'acquisition doit impérativement avoir une méthode `polling()` non nulle.

A réception d'un signal, la méthode `trap_sgn()` est exécutée, si elle existe. A elle d'effectuer le traitement correspondant au signal reçu.

5.3.5. Manipulation des tâches

Voici la structure de données définissant une tâche.

```
struct {
    TACHE_NOM    nom;
    TACHE_ETAT  etat;
    procedure    (*define)(),
                (*init)(),
                (*term)(),
                (*polling)(),
                (*traite)(),
                (*trap_sgn)( TACHE_NOM from,SGN signl );
} TACHES;
```

Il existe une variable globale `Self`, pointant sur la tâche active. Quand une tâche prend la main dans une de ses méthodes, elle peut utiliser `Self` pour se désigner elle-même et avoir ainsi accès à la structure de données `TACHES`. C'est d'ailleurs le seul moyen de faire. `Self` correspond exactement à `This` en C++.

```
- procedure define_tache( TACHE_NOM name,
                          procedure(*define_fct)(void));
```

Pour être connue par le superviseur, une tâche soit être définie. Ceci correspondant à la notion de constructeur en termes de programmation par objets. C'est dans cette procédure qu'une tâche donne la liste des méthodes qu'elle va utiliser.

```
- procedure signal( TACHE_NOM name,SGN signl );
```

Envoie le signal `signl` à la tâche `name`. La tâche qui reçoit le signal sait de quelle tâche il provient. Pour que ce signal soit utile, il faut que la tâche `name` ait prévu de récupérer ce signal (méthode `trap_sgn()` différente de `NULL`), sinon rien ne se passe. Lorsqu'une tâche envoie un signal, l'instruction qui suit immédiatement l'appel de `signal()` n'est exécutée qu'après la prise en compte du signal par la tâche recevante, c'est à dire après l'exécution de la méthode `trap_sgn()`.


```
- procedure atraiter();
```

Indique au superviseur que la tâche désire reprendre la main pour exécuter sa méthode `traite()`. Pour les tâches d'acquisition, nous avons admis que la seule opération effectuée dans la méthode `polling()` était d'acquérir les données, jusqu'à ce qu'une trame complète soit arrivée. Le traitement des données ne s'effectuant, par la méthode `traite()`, qu'une fois que les méthodes `polling()` de toutes les tâches auront été exécutées. Le superviseur lance ensuite l'exécution des méthodes `traite()`.

5.3.6. Les différents états d'une tâche

Voici le type `TACHE_ETAT`:

```
{ST_HS, ST_REPOS, ST_BLOQUE, ST_ENCOURS, ST_ERREUR};
```

La signification de ces cinq valeurs n'a pas besoin d'être commune à toutes les tâches. Cet indicateur n'est en effet utilisé qu'au sein même d'une tâche. Il est néanmoins préférable de respecter quelques règles pour une meilleure lecture du programme.

- **ST_HS** indique qu'une tâche n'est plus active. Cet état n'est utilisé que par la tâche `sched`, qui passe dans cet état sur réception du signal `SIGKILL`.
- **ST_REPOS** est par convention l'état d'une tâche qui vient d'être définie par `define_tache()`. En phase d'acquisition, cet état indique que la tâche n'est pas en dialogue avec un appareil distant.
- **ST_ENCOURS** n'est utilisé qu'en phase d'acquisition. Il indique qu'un dialogue est instauré avec l'appareil. Pour les appareils qui répondent à une demande du programme (Magnavox et thermosalinographe), la tâche passe dans cet état entre l'envoi de la demande et la réception de la réponse. Lorsqu'une tâche est dans cet état, elle doit refuser toute nouvelle demande d'acquisition, car celle-ci entrerait en conflit avec celle en cours. Le thermosalinographe par exemple n'est plus à l'écoute de l'ordinateur tant qu'il n'a pas envoyé sa réponse. Il serait donc de toute façon inutile d'essayer de dialoguer avec lui. Il indique soit un problème de transmission (la durée de transmission est plus longue que prévu), soit une période d'acquisition inférieure à la durée normale de réponse de l'appareil. Pour le thermosalinographe, il ne faut pas descendre en dessous de quinze secondes par exemple.
- **ST_BLOQUE** n'est utile qu'avec le multiplexeur. Il indique qu'une tâche n'arrive pas à s'approprier le multiplexeur (`rslock()` retourne `FALSE`). C'est un phénomène tout à fait normal dès qu'on utilise le multiplexeur. Il faut donc que les tâches soient capables de gérer cet état.
- **ST_ERREUR** indique une erreur quelconque (erreur de transmission, allocation mémoire etc...). Ceci se traduit normalement par l'envoi du signal `SIGERR`.

5.3.7. Description d'une tâche d'acquisition

Les tâches d'acquisition basées sur le module RS sont toutes écrites de la même façon. En voici les principes.

La première étape est la construction de l'objet tâche, c'est à dire l'initialisation de la structure de données pointée par `Self`. C'est la procédure `define_tache()`. La méthode `init()` permet d'installer les procédures de réception de données: initialisation du module RS, allocation mémoire éventuelle, etc... La méthode `term()` effectue la travail inverse de `init()`. Elle doit restaurer la mémoire allouée et fermer le port série.

Les méthodes `polling()` sont bâties sur le modèle suivant:

```
Si l'état de la tâche est ST_BLOQUE
essayer de s'approprier la liaison
si possible
    envoyer la demande de données à l'appareil
    armer le timeout
sinon
    rien a faire encore
sinon
    acquisition des données
    selon l'état de construction de la trame courante
        si fin de trame rencontrée
            désarmer le timeout
            appel de atraiter()
        si trame en cours, non complète
            si timeout atteint
                signaler l'erreur aux tâches ayant demandé l'acquisition
            sinon
                rien a faire encore
        si erreur sur les données reçues
            signaler l'erreur aux tâches ayant demandé l'acquisition
```

L'acquisition des données est basée sur un automate, dont les différents états indiquent l'état de réception des données. Ce principe est appliqué pour toutes les tâches d'acquisition. En voici l'algorithme succinct:

```
si aucun caractère n'a été reçu
    rien à faire encore
sinon
    selon l'état de l'automate
        etat 0 : /* premier caractère */
                /* recherche un paquet disponible */
                /* se met en etat d'attente des autres caractères */
            appel de new_trame()
            si ok, appel de reset_paquet()
            etat automate = 1
        etat 1 : /* la suite des données */
                /* stockage jusqu'à réception de la séquence */
                /* de fin de trame (<CR><LF> en général */
                /* test d'un éventuel débordement */
            selon le caractère reçu
                caractère quelconque:    appel de put_paquet()
                                           test débordement
                caractère de fin de trame: /* fin de trame */
                                           appel de send_appli()
```

Une tâche d'acquisition doit s'attendre à recevoir le signal SIGASK. Ceci doit normalement déclencher l'envoi de la demande de données à l'appareil (gestion possible d'un blocage si utilisation du multiplexeur -> etat=ST_BLOQUE). On doit refuser d'envoyer cette demande si un dialogue est déjà en cours (etat=ST_ENCOURS). Si la demande peut être effectuée, passage dans l'état ST_ENCOURS.

La dernière méthode décrite ici est `traite()`. Elle est chargée de l'analyse et du traitement des données accumulées lors des appels de `polling()`. L'exécution de cette méthode est déclenchée par l'appel de `atraiter()` (voir plus haut). Le contenu de cette méthode dépend fortement des données reçues, donc de l'appareil considéré, mais quelques principes restent constants:

```
Si l'appareil peut envoyer plusieurs types de trames
    identifier la trame reçue
analyse de la trame
détection d'éventuelles erreurs
stockage des données analysées dans une zone mémoire protégée, propre
à la tâche
remise du paquet contenant les données brutes dans le pool des paquets
disponibles (appel de lib_paquet()).
Si l'ensemble des données a été reçu,
    /* cas de données réparties dans plusieurs trames */
    libération de la liaison par l'appel de rsunlock()
    /* important en cas d'utilisation du multiplexeur */
```

5.3.8. Description d'une tâche de stockage de données

La tâche la plus importante dans un programme d'acquisition de données est celle permettant l'écriture sur disque. Bien qu'il soit plus difficile de décrire un fonctionnement typique pour cette tâche que pour une tâche d'acquisition, en voici une description simplifiée. La méthode la plus importante ici n'est plus `polling()`, mais `trap_sgn()`. Le seul rôle de `polling()` est de déclencher les acquisitions des appareils lorsque le compteur d'acquisition arrive à 0. On suppose que le compteur (compte à rebours) est lancé la première fois par la méthode `init()`. Par contre, c'est `trap_sgn()` qui va recevoir les réponses des différentes tâches d'acquisition, au fur et à mesure des arrivées de données. En cas de fonctionnement du programme sans multiplexeur, toutes les acquisitions se font simultanément et les réponses arrivent dans n'importe quel ordre, suivant la vitesse à laquelle l'appareil de mesure ou de navigation est capable de les calculer et de les envoyer. La plus grosse difficulté est donc d'attendre que toutes les tâches d'acquisition aient répondu avant de déclencher la sauvegarde. Lorsqu'on utilise un récepteur GPS de type NMEA-0183 ou NALNO, une autre difficulté apparaît, liée au fait que ces récepteurs envoient la position du navire périodiquement, sans qu'aucune demande ne leur soit envoyée. Il faut donc être capable de filtrer parmi toutes ces données celles que l'on retiendra pour le stockage (en général, la première arrivée dès la période d'acquisition écoulée).

Algorithme de `polling()` :

```
Si le compteur d'acquisition est claqué
    lancer les demandes d'acquisition à toutes les tâches
    définies comme tâches d'acquisition.
    Construire la liste de ces tâches
    réarmer le compteur d'acquisition
    Passer de l'état ST_REPOS à ST_ENCOURS
```

Algorithme de trap_sgn():

Selon le signal reçu

SIGSTART : Envoyé par le superviseur
C'est ici que l'on écrit dans le fichier de données pour indiquer un début d'acquisition

SIGTERM : Envoyé par le superviseur
C'est ici que l'on écrit dans le fichier de données pour indiquer une fin d'acquisition

SIGASW : Envoyé par une tâche liée à un appareil de mesures
Si l'état de la tâche n'est pas ST_ENCOURS
ne rien faire
sinon
Retirer la tâche de la liste construite par polling()
Si la liste est vide /* On a reçu toutes les réponses */
stockage des données
retour à l'état ST_REPOS
sinon
rien à faire encore

SIGGPS :
SIGEST :
SIGFIX :
SIGSAT : Envoyé par une tâche liée à un appareil de navigation
Traitement identique à SIGASW.

5.3.9. Les outils de mise au point

5.3.9.1. Le mode trace

Le programme THERMO offre la possibilité d'afficher dans une fenêtre tous les caractères échangés entre l'ordinateur et un appareil. On ne peut tracer qu'un seul appareil à la fois. Le choix se fait par le passage d'un paramètre sur la ligne de commande: "thermo -tn" pour tracer la navigation ou "thermo -tt" pour tracer le thermosalinographe. Cette option est typiquement utilisée lors de la mise au point de la connexion d'un nouvel appareil et son intégration dans THERMO. Cette option peut être néanmoins utile pour surveiller les échanges entre un appareil et l'ordinateur. On peut ainsi détecter plus facilement un problème quelconque dans le dialogue.

5.3.9.2. Le mode simulation

Le mode simulation permet d'exécuter le programme sans être obligé d'effectuer réellement la connexion des divers appareils. C'est un outil de développement très important, car il permet de mettre au point le programme sans pour autant immobiliser un récepteur satellite et un thermosalinographe. Dans le programme THERMO, chaque appareil intégré au logiciel existe avec son module de simulation. L'appel se fait par "thermo -st" ou "thermo -sn" selon l'appareil que l'on veut simuler.

Les fonctions de simulation remplacent les fonctions du niveau 1 (RS).

6. L'UTILISATION DE SF DANS LE PROGRAMME

6.1. Les principes généraux de SF

SF est une bibliothèque logicielle, écrite en langage C (Turbo-C sur IBM-PC, Unix-C sur SUN) constituée de deux modules principaux: une bibliothèque de gestion d'écrans qui permet l'affichage de données dans une ou plusieurs fenêtres, leur modification de façon interactive ou à travers un programme et une bibliothèque de gestion de fichiers qui permet le stockage et la relecture des données dans des fichiers ASCII.

L'originalité de ce logiciel tient dans le fait que la description des données que l'on veut manipuler est complètement extérieure au logiciel lui-même. Celle-ci se fait par la donnée de **descripteurs d'écrans** et de **descripteurs de fichiers**. Ces descripteurs sont des fichiers texte, donc facilement modifiables, dont le rôle est de déterminer divers paramètres permettant de configurer le comportement du logiciel. Ainsi, l'emplacement et la taille d'une fenêtre d'affichage, l'emplacement et le format d'affichage des données dans cette fenêtre sont des paramètres se trouvant entre autres dans un descripteur d'écrans. Un descripteur de fichiers contient le format de stockage des données sur disque (longueur et type). Chaque donnée fait partie d'un **enregistrement**. Un enregistrement est stocké obligatoirement sur une ligne, mais un même fichier peut contenir plusieurs types d'enregistrements différents.

SF fournit un ensemble de fonctions élémentaires permettant de créer et d'accéder à ces données, de les modifier, soit de façon interactive par une saisie manuelle quand il s'agit de stocker des données dont le support n'est pas informatisé, soit directement par un programme lorsque cela est possible.

SF peut donc convenir à des applications aussi diverses que par exemple la saisie manuelle de données, le stockage de données acquises automatiquement ou la gestion *agréable* des paramètres de configuration d'un logiciel.

Un atout intéressant, lié à la définition extérieure à un programme des fenêtres d'affichage, est qu'il est très facile d'adapter une version en langue étrangère d'un logiciel conçu autour de SF.

De plus, SF est un système très évolutif. Lorsque l'on écrit un logiciel construit autour de SF, il est très facile de modifier la structure des masques ou des fichiers gérés par SF. On n'a, en effet, pas à se préoccuper de l'organisation réelle des données sur disque ou dans un masque, l'accès à une **variable** dans un masque s'effectue de la même manière quelque soit le nombre total de variables dans ce masque. Il en va de même pour les variables de fichier. Il est donc très facile d'ajouter une variable par exemple, car on sait que cela ne perturbera pas le reste du programme. C'est un atout non négligeable quand il s'agit de faire évoluer un logiciel rapidement, comme ce fut le cas de THERMO.

La définition complète des descripteurs d'écrans et de fichiers est fournie en annexe de ce document (Buisson, 1989).

6.2. La bibliothèque de gestion d'écrans

6.2.1. Description

La bibliothèque **SF-Gestion d'écrans** fournit l'ensemble des primitives nécessaires à l'affichage et la modification de données de tous types (entiers, réels, chaînes de caractères). Les données que l'on veut manipuler sont regroupées en un ou plusieurs **masques** selon des critères propres à chaque application. Chacun de ces masques, décrit par un descripteur d'écran, est affiché à l'intérieur d'une fenêtre. Lors d'une saisie manuelle d'un tel masque, toutes sortes de déplacements sont admis: de caractère en caractère à l'intérieur d'un même champ de saisie (*flèche gauche, flèche droite*), de champ en champ (*flèche bas, flèche haut*), retour direct sur le premier champ (*touche home*), accès direct sur le dernier champ (*touche end*). Chaque champ de saisie peut être protégé par une **formule de contrôle** lors de chaque saisie. Toute valeur saisie sera refusée si elle ne satisfait pas cette formule. La validation de toutes les données d'un masque de saisie se fait par la touche **F2**. L'abandon est obtenu par la touche **ESC**.

6.2.2. Les fonctions d'interface

L'utilisation de la bibliothèque SF-Gestion d'écrans dans un programme nécessite la connaissance des primitives suivantes:

- `MASQUE mopen(char *filename);`

`mopen()` est la première fonction à utiliser. Elle crée en mémoire une structure de type `MASQUE`, image du descripteur d'écran `filename`.

- `procedure maffiche(MASQUE masque);`

`maffiche()` permet l'affichage d'un masque préalablement chargé en mémoire par `mopen()`. Cette procédure ouvre une fenêtre à l'écran et affiche le contenu du masque.

- `procedure mrefresh(MASQUE masque);`

`mrefresh()` est une procédure qui n'affiche que les valeurs des variables contenues dans le masque. C'est une fonction très utile lorsque l'on veut entretenir des valeurs à l'écran au fur et à mesure de leur évolution. `maffiche()` doit cependant avoir été appelée au moins une fois avant pour ouvrir la fenêtre.

- `booleen msaisit(MASQUE masque);`

`msaisit()` permet l'affichage et la saisie manuelle d'un masque. L'utilisateur peut saisir ou modifier les valeurs préalablement saisies contenues dans ce masque. Il existe deux possibilités pour sortir de cette fonction. La touche **F2** permet de valider toutes les modifications éventuelles, **ESC** permet de les annuler. Le code de retour de cette fonction est `VALIDE` pour **F2** et `ABORT` pour **ESC**. Il est du ressort du programme appelant de traiter ces codes de retour selon ses propres besoins.

`msaisit()` peut également être utilisée pour saisir un menu. La sélection se fait en déplaçant la barre en inversion vidéo jusque sur le choix voulu (*Flèche-haut/Flèche-bas*) puis en tapant *Return* ou

directement par l'appui d'une touche de fonction ($F1$ à Fn selon le nombre de choix possibles). Dans ce cas, `msaisit()` retourne le numéro du choix effectué (de 0 à $n-1$). La touche `ESC` est toujours disponible dans un menu et génère le code de retour `-1`. Il est de la responsabilité du programme appelant de gérer ces codes.

Remarque: La syntaxe d'un descripteur d'écran diffère selon que l'on décrit un masque de saisie ou un menu (Buisson, 1989).

- `char *mget (MASQUE masque, char *varname);`

Dans un masque, chaque valeur est repérée par un nom (chaîne de 8 caractères max). On peut récupérer à tout moment la valeur associée à une variable dans un masque. Cette valeur est retournée par `mget()` sous forme d'une chaîne de caractères, quelque soit le type réel de la variable; exemple: si une variable contient la valeur entière 123, `mget()` retournera la chaîne de caractères "123".

- `booleen mput (MASQUE masque, char *varname, char *valeur);`

De façon symétrique, on peut associer par programme une valeur à n'importe quelle variable d'un masque. `mput()` retourne `TRUE` si la variable de nom `varname` est bien trouvée dans le masque et `FALSE` dans le cas contraire. De même que pour `mget()`, la valeur que l'on veut stocker dans le masque doit être une chaîne de caractères; exemple: pour stocker la valeur entière 123, il faut passer le paramètre "123" à `mput()`. La conversion dans le type réel de la variable s'effectue et la valeur est tronquée si celle-ci est de taille supérieure à la place réservée dans le descripteur d'écran.

- `procedure mclose (MASQUE masque);`

`mclose()` est la procédure qui permet de restaurer la mémoire après l'utilisation d'un masque. Cette procédure doit être appelée sur chaque masque ouvert par `mopen()`.

- `procedure tinit();`

Procédure d'initialisation de la bibliothèque. Doit être appelée avant toute autre.

- `procedure tterm();`

Procédure de sortie de la bibliothèque. Doit être appelée en fin de programme.

6.3. La bibliothèque de gestion de fichiers

6.3.1. Description

SF-Gestion de fichiers contient les primitives permettant la création d'un fichier ASCII selon un format spécifié dans un **descripteur de fichier** associé, ainsi que toutes les opérations de relecture, de modification, de suppression ou d'ajout d'un ou plusieurs enregistrements. Lorsque ces opérations doivent se faire de façon interactive, il est indispensable d'associer un **masque de saisie** à chaque type d'enregistrement. L'ensemble de ces masques permet d'effectuer les modifications voulues sur une ou plusieurs valeurs, modifications

répercutées ensuite au niveau du fichier de données. Lors du parcours d'un fichier de données, les principaux déplacements autorisés sont les suivants: enregistrement suivant: *PgDn*, enregistrement précédent: *PgUp*, premier enregistrement: *Ctrl-PgUp*, dernier enregistrement: *Ctrl-PgDn*, en plus des déplacements à l'intérieur d'un masque. A tout moment, la touche *F2* permet de valider l'ensemble du fichier alors que *ESC* permet d'abandonner les éventuelles modifications.

6.3.2. Les fonctions d'interface

La bibliothèque SF-Gestion de fichiers est bien sûr basée sur SF-Gestion d'écrans. Les primitives suivantes permettent de gérer les fonctions de stockage et de lecture disque.

```
- DISQUE dopen( char *filename, char *dskname );
```

`dopen()` est la première procédure à utiliser. Elle permet d'associer un descripteur de fichier à un fichier de données, existant ou non. Cette procédure ne lit pas les données, mais construit simplement en mémoire la structure de données de type `DISQUE`, utilisée dans toute la suite, à l'image de `dskname`. Lorsque des masques de saisie sont associés à des enregistrements du fichier, tous les `mopen()` correspondants sont effectués.

```
- procedure dcreate( DISQUE disk );
```

La création d'un fichier de données est assurée par cette procédure. Le mode de saisie des données est interactif. Chaque enregistrement est saisi avec le masque de saisie qui lui est associé.

```
- procedure dload( DISQUE disk );
```

La lecture du fichier de données est effectuée par cette procédure. Cette procédure ne peut être utilisée que sur un fichier déjà existant. Les données du fichier sont lues conformément au format donné dans le descripteur de fichier et stockées entièrement en mémoire. Ce choix a été fait pour optimiser au maximum les accès aux données, sachant qu'en contre partie, la taille maximale d'un fichier est donnée par la taille de la mémoire conventionnelle disponible (640 Ko moins le système et les programmes résidents sur un IBM-PC).

```
- procedure dsave( DISQUE disk );
```

Cette procédure est l'inverse de la précédente: elle sauvegarde sur disque le contenu de la mémoire centrale.

```
- boolean dedit( DISQUE disk, INSTANCE instance );
```

`dedit()` permet le parcours et la modification de l'ensemble des données contenues dans un fichier. Chaque enregistrement de données, une fois chargé en mémoire par `dload()`, est stocké dans une structure de données de type `INSTANCE`. Le paramètre de la procédure indique l'enregistrement à partir duquel on veut commencer le parcours (en général le premier du fichier, mais ce n'est pas une obligation). Il existe encore deux moyens de sortir de cette procédure: *F2* donne un code de retour `TRUE` et *ESC* un code `FALSE`. Il est de la responsabilité du programme appelant d'effectuer les traitements adéquats dans ces deux cas.


```
- INSTANCE first_instance( struct t_arbre *arbre );
```

first_instance() retourne la première instance (premier enregistrement) du fichier. C'est typiquement la fonction qu'il faut appeler comme deuxième paramètre de dedit().

Exemple d'appel:

```
disk = dopen( "données","descripteur" );  
dedit( disk,first_instance( disk->arbre ) );
```

```
- INSTANCE dpos( struct t_arbre *arbre,char *inst_name,  
                unsigned char numéro );
```

On peut avec la fonction dpos() se positionner sur l'instance numéro numéro de type inst_name. Cette fonction est également un deuxième paramètre possible pour dedit(). La numérotation des instances commence à 0. Il existe également deux codes spéciaux permettant de se positionner sur la première instance: D_FIRST ou sur la dernière: D_LAST, toujours pour un type d'instance donné.

```
- char *dget( INSTANCE instance,char *varname );
```

On peut récupérer la valeur d'une variable dans une instance avec dget(). Là encore, la valeur retournée est une chaîne de caractères.

```
- boolean dput(INSTANCE instance,char *varname,char *valeur);
```

On peut stocker une valeur dans une variable d'instance par dput(). La valeur doit être une chaîne de caractères, qui sera convertie et tronquée au besoin.

```
- procedure dclose( DISQUE disk );
```

dclose() doit être appelée en fin de programme pour libérer la mémoire utilisée après un dopen(). Les mclose() correspondants aux éventuels masques de saisie sont effectués à ce moment là. Après un dclose(), toutes les informations stockées en mémoire sont perdues, il faut donc sauver sur disque les éventuelles modifications par un dsave().

6.4. SF dans le programme THERMO

Dans le programme THERMO, SF n'a pas été utilisé pour le stockage de données. En effet, ce programme se contente de stocker les données acquises sur disque, aucun traitement n'est effectué. Il était donc inutile d'utiliser SF alors qu'une fonction standard en langage C d'écriture sur disque suffit. Par contre, SF nous a permis dans ce programme de créer le menu principal, d'afficher à l'écran les données acquises et d'avoir un moyen de configurer complètement le logiciel de façon simple, rapide et sans re-saisie inutile des paramètres qui évoluent peu.

6.4.1. Configuration

Les paramètres de configuration de THERMO sont stockés dans un fichier ASCII qui est accédé à travers SF-Gestion de fichiers. Ce fichier (`config.cfg`) est constitué de deux types d'enregistrements différents:

- Paramètres généraux du logiciel: ce sont les paramètres que l'on peut être amené à modifier souvent. On y trouve le nom du fichier de sauvegarde des données, le choix du navigateur par satellite, la période l'échantillonnage et un indicateur de présence du multiplexeur de liaisons série.
- Configuration des liaisons avec les appareils. Tous les appareils utilisés dans cette application sont connectés par une liaison série à l'ordinateur. Il faut donc configurer les paramètres de communication (vitesse de transmission, nombre de bits de données et de stop, type de parité). Il existe dans le fichier un enregistrement par appareil, contenant toutes ces informations. Celles-ci ne doivent normalement pas être modifiées une fois le logiciel installé correctement.

Il existe un deuxième fichier de configuration (`constant.cfg`), propre cette fois au thermosalinographe, contenant les coefficients d'étalonnage de l'appareil utilisé.

6.4.2. Saisie de la position manuelle

Le programme THERMO peut être utilisé sans système de navigation automatique ou lorsque celui-ci tombe en panne. Il faut dans ce cas saisir la position manuellement. Dans la phase d'acquisition, la touche `Ctrl-F10` permet l'ouverture d'un masque de saisie. L'utilisateur peut alors saisir la position et l'heure GMT. Ces informations sont affichées à l'écran et stockées sur disque.

6.4.3. Présentation des données

Nous avons associé à chaque appareil une fenêtre dans laquelle vient se faire l'affichage des données de cet appareil. Sur l'écran, on peut alors visualiser en temps réel, au fur et à mesure des acquisitions les données du thermosalinographe et les données de navigation.

En plus de ces données, nous présentons quelques informations concernant l'état du programme. Il est très important de savoir à tout moment si tout ce passe bien, pour ne pas risquer de perdre des données si un problème se pose. Le programme affiche donc continuellement dans une fenêtre l'état du dialogue avec les différents appareils connectés (demande en cours, réponse reçue, erreur). Avec les récepteurs

satellites que nous utilisons, nous savons également si nous sommes en couverture GPS ou non. Dans ce cas là, une fenêtre indique la perte de GPS. On peut alors facilement savoir grâce aux moyens de navigation du bord s'il s'agit d'une panne du récepteur ou bien d'une absence de couverture GPS.

Le programme affiche enfin l'heure de l'ordinateur, qui doit être l'heure GMT pour être facilement comparée à l'heure du satellite, et un compteur à rebours indiquant combien de temps il reste avant la prochaine acquisition. On peut donc voir d'un coup d'oeil que tout se passe bien en regardant ce compteur. Si l'horloge est figée, on sait également depuis combien de temps, ce qui peut faciliter l'analyse des données par la suite en marquant la durée de l'interruption.

7. UTILISATION DU PROGRAMME THERMO

Ce chapitre décrit l'utilisation du système de mesure de température et de salinité de surface installé sur des bateaux marchands. Il est tiré d'un document en Français distribué aux officiers chargés de la surveillance de l'équipement. Cette documentation, ainsi que le logiciel existent en version Anglaise (Grelet al, 1992).

7.1. Principes

Le système d'acquisition automatique de température et de salinité de surface permet d'échantillonner ces deux paramètres en continu sur des bateaux marchands.

Un thermosalinographe, de marque Sea-Bird, modèle SBE21, est branché sur le système de refroidissement de la machine et transmet les données à un micro-ordinateur installé à la passerelle.

Le thermosalinographe possède 4 vannes qui doivent être branchées comme suit:

- La sortie **OUT** est connectée sur la vanne de coque ou tout autre sortie basse pression.
- L'entrée **SALT WATER IN** est connectée sur une prise d'eau utilisée pour le refroidissement de la machine. Cette prise d'eau doit se trouver le plus près possible du circuit de pompage.
- L'entrée **FRESH WATER** doit être connectée à une prise d'eau douce délivrant au moins 2 bars afin d'assurer un rinçage correct de l'appareil.
- La sortie **DRAIN** est utilisée pour vidanger le thermosalinographe. L'évacuation peut donc se faire directement dans la cale du bateau.

Toutes les connexions sont réalisées dans du tube PVC, il est donc important de ne pas dépasser la pression maximale autorisée pour ce type de matériaux: **50 PSI maximum (3.5 bars)**.

Le thermosalinographe est relié au micro-ordinateur par l'intermédiaire d'un câble étanche 4 conducteurs et d'une boîte de jonction. Cette boîte de jonction est placée à la passerelle et permet l'alimentation électrique du thermosalinographe et une transmission opto-électronique des données.

Un système de navigation autonome GPS, modèle Pronav GPS100, peut être connecté au micro-ordinateur pour une acquisition automatique de la position.

Un multiplexeur de lignes RS232 est utilisé conjointement avec un micro-ordinateur portable afin de disposer de quatre liaisons série. Dans la configuration standard (thermosalinographe et récepteur GPS), seuls deux voies sont utilisées.

Les données sont acquises en continu et stockées toutes les 5 minutes sur une disquette 3 1/2 pouces.

Le groupe SURTROPAC de nouméa se charge de l'entretien du système à chaque escale du navire. Les disquettes de données sont vérifiées et remplacées par de nouvelles. Le fonctionnement du programme est testé. Le thermosalinographe est démonté, nettoyé au laboratoire et remplacé si nécessaire.

7.2. Mise en route du thermosalinographe

Les vannes doivent être positionnées comme indiqué dans le tableau ci-dessous en fonction de l'utilisation souhaitée. Une vanne est ouverte lorsque sa poignée de commande est horizontale. Avant chaque arrivée dans un port, il est demandé au personnel de la machine, de rincer le système avec de l'eau douce pendant plusieurs minutes afin d'éviter tout dépôt de sels sur les électrodes du capteur de conductivité.

| OPERATION | VANNE | FONCTION | POSITION |
|--|--|--|--|
| Utilisation normale | Sortie Entrée d'eau douce Entrée d'eau de mer Vidange | Ouverte Fermée Ouverte Fermée | Horizontale Verticale Horizontale Verticale |
| Rinçage eau douce (à faire entre chaque période d'utilisation) | Sortie Entrée d'eau douce Entrée d'eau de mer Vidange | Ouverte Ouverte Fermée Fermée | Horizontale Horizontale Verticale Verticale |
| Stockage (ou électronique enlevée pour étalonnage) | Sortie Entrée d'eau douce Entrée d'eau de mer Vidange | Fermée Fermée Fermée Fermée | Verticale Verticale Verticale Verticale |

NOTE: PRESSION D'UTILISATION MAXIMUM = 50 PSI (3.5 bars)

Afin de pouvoir vérifier le bon fonctionnement de l'appareil, il est conseillé d'échantillonner de l'eau de mer une fois par jour. Ce prélèvement est obtenu en ouvrant la vanne de purge de l'appareil (drain). Ces échantillons seront ensuite récupérés au retour du bateau et analysés au laboratoire.

7.3. Utilisation des écrans dans le programme

Terminologie

Le programme fait un usage fréquent des touches du pavé numérique. Pour simplifier l'écriture nous utiliserons les mots suivants:

- **PageHaut** pour l'appui sur la touche PageUp.
- **PageBas** pour l'appui sur la touche PageDown.
- **Haut** pour l'appui sur la touche flèche Up.
- **Bas** pour l'appui sur la touche flèche Down.
- **F1** pour l'appui sur la touche flèche de fonction F1 par exemple.
- **Ctrl-F1** pour l'appui simultané sur les touches spéciales Control et F1.
- **ESC** pour l'appui sur la touche de sortie Escape.
- **Return** pour l'appui sur la touche de validation Retour chariot.

Le programme **THERMO** utilise 3 types de fenêtres:

- la fenêtre **menu**.
- la fenêtre **édition**.
- la fenêtre **affichage**.

La fenêtre **menu** est utilisée pour modifier la configuration du système. Ce menu n'est accessible que lorsque l'option (-m) à été spécifiée sur la ligne de commande. Lorsque le menu apparaît, l'utilisateur peut faire son choix de différentes manières:

1. En appuyant sur la touche de fonction correspondant à l'option choisie.
2. En utilisant les touches **Haut** et **Bas** du pavé numérique pour se déplacer sur l'option choisie, puis en validant par l'appui sur la touche **Return**
3. L'appui sur la touche **ESC** permet de sortir du programme et de retourner sous **DOS**.

La fenêtre **edit** est utilisée pour modifier la configuration du système ou par exemple, entrer manuellement la position du navire .

1. Les touches **Haut** et **Bas** du pavé numérique permettent de se déplacer dans le masque de saisie d'un champ à un autre. Choisir le champ à modifier et entrer sa valeur. Cette dernière est validée par l'appui sur les touches **Return**, **Haut** ou **Bas**. Une entrée incorrecte est refusée et signalée par un signal sonore.
2. Si le programme le permet, d'autres fenêtres de configuration sont accessibles par l'appui successif sur la touche **PageBas**. La touche **PageHaut** permet de revenir aux fenêtres précédentes.
3. Frappez **F2** si vous voulez revenir au menu principal tout en sauvegardant les données précédemment modifiées dans le fichier de configuration.
4. Frappez **ESC** si vous ne voulez pas modifier le fichier de configuration. Le programme vous pose alors la question : "Sortie sans écriture disque O/N ?". Entrer la bonne réponse. en frappant la lettre **O** ou **N**.

Les fenêtres **d'affichage** sont utilisées uniquement pour l'affichage des données à l'écran (position GPS, réponse du thermosalinographe, dernière position du navire, etc ...).

7.4. Utilisation du système

Ce système est prévu à l'origine pour fonctionner avec différentes configurations matérielles. Cette configuration est faite par le technicien de maintenance et ne doit pas être modifiée à bord du navire. Habituellement, sur les navires marchands, le système de navigation par satellite est un récepteur Pronav GPS100 (NMEA-0183). Le thermosalinographe et le GPS sont connectés au micro-ordinateur Zenith par l'intermédiaire d'un multiplexeur de voies série. Cette configuration est utilisée pour décrire le fonctionnement du logiciel.

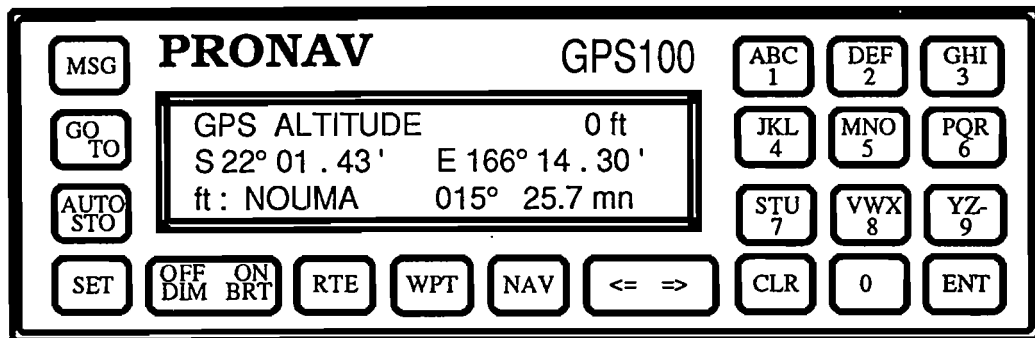


Fig 2. Panneau avant du récepteur Pronav GPS100.

Mise en route:

1. Insérer la disquette programme dans le lecteur A (stick rouge).
2. Insérer la disquette données dans le lecteur B (stick bleu).
3. Mettre en route la boîte de jonction Sea Bird et le multiplexeur en mettant les interrupteurs sur ON. Le voyant rouge de mise sous tension doit s'allumer.
4. Mettre sous tension le récepteur Pronav GPS100 en maintenant appuyée la touche **ON**, la page relative aux tests internes est affichée pendant environ deux secondes. Ensuite la page d'initialisation est affichée. Elle permet de vérifier, et si nécessaire, de modifier la date et l'heure GMT. Appuyer ensuite sur la touche **ENTER** et le récepteur commence l'acquisition des satellites. Ce processus est automatique et prend environ 2 à 3 minutes pour fournir une information de position, (cf Manuel d'utilisation, Pronav, 1990).
5. Mettre en route le micro-ordinateur portable et le laisser en fonctionnement pendant la totalité du voyage. Il est conseillé de baisser l'intensité lumineuse de l'écran la nuit. Il est possible d'éteindre uniquement l'écran sur les micro-ordinateurs portables Zenith en appuyant simultanément sur les touches **FN** et **F10**. Un nouvel appui sur ces touches allumera l'écran de nouveau.
6. Le programme est automatiquement chargé en mémoire et l'acquisition des données commence. Il est conseillé de vérifier après chaque nouveau démarrage le bon fonctionnement du programme. Après quelques secondes, lorsque les réponses des différents appareils ont été correctement analysées, l'écran suivant apparaît:

| Orstom Nouméa - B.Buisson - J.Grelet - V4.8 Nov 91 | | |
|--|---|--------------------|
| Données Navigation GPS (GP188 NMEA 183) | | |
| Date : 13/ 5/92 | Heure GMT : 12.49 | THERMOSALINOGRAPHE |
| Latitude : +1.38° | Longitude : -166.16° | REPONSE recue |
| SOG : 12.3 Mds | COG : 388.8° | OK |
| | | NMEA 0183 |
| | | REPONSE recue |
| | | OK |
| | | DISQUE |
| | | Reste 340 Ko |
| | | sur disque B: |
| | | soit 48% libre |
| Acquisition Thermosalinographe SBE21 | | |
| Jour Julien : 134 | Heure GMT : 12.49 | Mb cycles : 2 |
| Température : 31.255 | Conductivité : 5.971 | Salinité : 35.012 |
| Il est : 12:49:43 | Prochaine acquisition dans 00004 secondes | |
| Sortie : Ctrl-F1. | Position : Ctrl-F10. | |

Fig 3. Aspect de l'écran au cours de l'acquisition.

En phase d'acquisition, la fenêtre **Données Navigation GPS** est rafraîchie toutes les deux secondes et la fenêtre **Acquisition Thermosalinographe SBE21** au rythme choisi par l'utilisateur. En règle générale toutes les cinq minutes.

7. L'opérateur peut à tout instant entrer manuellement la position du navire en appuyant sur les touches **Ctrl-F10**. Voir l'étape n°11. Il est important de ne pas laisser la fenêtre ouverte trop longtemps car l'acquisition des données est alors interrompue. Dès la saisie terminée, refermer le masque en appuyant sur **F2**.
8. La fenêtre **DISQUE** est affichée lorsque l'option sauvegarde est active. Elle donne la taille disponible sur le disque ou le lecteur de disquette où sont stockées les données, habituellement "b:" et le pourcentage de place encore disponible. Il est conseillé de changer de disquette lorsque cette valeur devient inférieure à 10%.
9. L'arrêt du programme se fait en appuyant sur les touches **Ctrl-F1**.
10. Pour redémarrer l'acquisition, trois possibilités:
 - Taper sur le clavier **thermo** suivi de l'appui sur la touche **Return**.
 - Appuyer simultanément sur les touches **Alt Ctrl Del** pour ré-initialiser l'ordinateur.
 - Reprendre le déroulement des opérations (voir l'étape n°5), en éteignant puis en rallumant l'ordinateur.
11. Si le système ne possède pas de récepteur satellite, l'opérateur doit entrer la position toutes les deux heures au format suivant:
 - Latitude: DD MM H où DD représente les degrés, MM les minutes et H l'hémisphère.
 $0 \leq DD \leq 90$
 $0 \leq MM \leq 59$
 H = S pour sud, H = N pour nord
 - Longitude: DDD MM H où DDD représente les degrés, MM les minutes et H le méridien.

$0 \leq DDD \leq 180$

$0 \leq MM \leq 59$

H = E pour est, H = O = W pour ouest

- Jour julien: JJJ représente le nombre de jours écoulés dans l'année depuis le premier janvier. Un tableau en annexe A donne le calendrier des jours julien.
- L'heure: HH MM où HH représente l'heure et MM les minutes.

| Orstom Nouméa - B.Buisson - J.Grelet - U4.8 Nov 91 | | |
|---|---|---|
| Données Navigation GPS (GP188 NMEA 183) | | |
| Date : 13/ 5/92 | Heure GMT : 12.58 | THERMOSALINOGRAPHE |
| Latitude : +1.38° | Longitude : -166.16° | DEMANDE en cours |
| SOG : 12.3 Nds | COG : 388.8° | NMEA 0183 |
| Position du navire à saisir manuellement par l'officier | | REPONSE recue |
| Latitude : 1 38 N (dd mm N/S) | Longitude : 166 16 W (ddd mm E/W) | OK |
| Jour Julien : 134 (JJJ) | Heure (GMT) : 12 58 (hh mm) | DISQUE |
| Validation F2 | | Reste 348 Ko sur disque B: soit 48% libre |
| Acquisition Thermosalinographe SBE21 | | |
| Jour Julien : 134 | Heure GMT : 12.49 | Nb cycles : 2 |
| Température : 31.255 | Conductivité : 5.971 | Salinité : 35.812 |
| Il est : 12:58:18 | Prochaine acquisition dans 88013 secondes | |
| Sortie : Ctrl-F1. | Position : Ctrl-F18. | |

Fig 4.Saisie manuelle de la position par l'opérateur.

Lorsqu'un champ est rempli, le passage au suivant se fait par l'appui sur la touche **Bas** ou **Return**. Un test de validité est effectué. Si la valeur est mauvaise elle est rejetée et un signal sonore est émis. L'opérateur, une fois les données saisies, peut se déplacer dans le masque avec les touches **Haut** et **Bas**. Lorsque toutes les données sont correctement entrées, la validation se fait par l'appui sur **F2**.

12. Une fois les données validées, elles sont sauvegardées sur disque et la fenêtre **Dernière position du navire** est affichée à l'écran. Son contenu restera identique tant qu'une nouvelle position n'aura pas été saisie.

| Orstom Nouméa - B.Buisson - J.Grelet - U4.8 Nov 91 | | |
|--|---|---|
| Données Navigation GPS (GP188 NMEA 183) | | |
| Date : 13/ 5/92 | Heure GMT : 12.52 | THERMOSALINOGRAPHE |
| Latitude : +1.38° | Longitude : -166.16° | REPONSE recue |
| SOG : 12.3 Nds | COG : 388.8° | OK |
| Derniere position du navire | | NMEA 0183 |
| Jour saisie : 134 | Heure GMT saisie : 12.58 | REPONSE recue |
| Jour système : 134 | Heure GMT système : 12.51 | OK |
| Latitude : +1.38° | Longitude : -166.16° | DISQUE |
| saisie par l'officier | | Reste 348 Ko sur disque B: soit 48% libre |
| Acquisition Thermosalinographe SBE21 | | |
| Jour Julien : 134 | Heure GMT : 12.52 | Nb cycles : 7 |
| Température : 31.255 | Conductivité : 5.971 | Salinité : 35.812 |
| Il est : 12:52:21 | Prochaine acquisition dans 88884 secondes | |
| Sortie : Ctrl-F1. | Position : Ctrl-F18. | |

Fig 5. Affichage de la position saisie manuellement par l'opérateur.

Dans le cas où l'opérateur ne peut terminer la saisie de la position, il est toujours possible de rendre la main au programme en appuyant sur la touche **ESC**.

7.5. Options de la ligne de commande

Toutes les options de la ligne de commande doivent commencer par un tiret (-) et doivent être séparées les unes des autres et de la commande thermo par au moins un espace.

Le format générique de la ligne de commande est le suivant:

```
thermo [-option] [-option]
      -m           : menu
      -t [device]  : mode trace
      -s [device]  : simulation
      t            : thermosalinographe
      n            : appareil de navigation
```

Si vous tapez seulement thermo suivi de **Return**, le programme se charge en mémoire et utilise les options par défaut.

Affichage du menu option (-m)

13. Cette option provoque l'affichage du menu à l'écran.

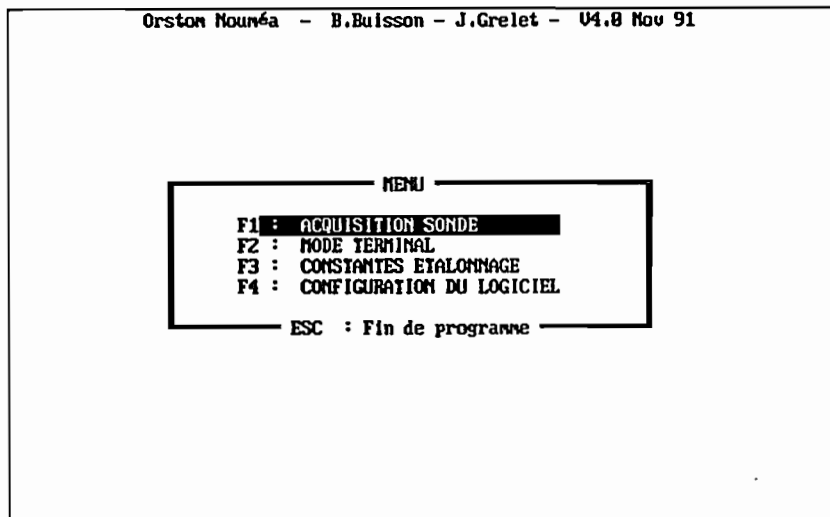


Fig 6. Menu principal.

14. Pour démarrer l'acquisition, sélectionner l'option **F1** et se reporter à la description numéro 6.
15. Le mode terminal, sélectionné par **F2**, permet d'instaurer un dialogue entre l'utilisateur et le système distant branché sur les ports série COM1 ou COM2. L'ordinateur se comporte alors comme un terminal ASCII. Ce mode est très utile pour tester la réponse des appareils connectés et peut remplacer de nombreux logiciels de communication du commerce.

L'appui sur les touches **F1** ou **F2** permet de passer de COM1 à COM2 ou de la voie1 à la voie2 si le multiplexeur est utilisé. Les différents paramètres de communication (vitesse, parité, etc...)

sont affichés en haut de l'écran. La sortie du mode terminal se fait par l'appui sur la touche **ESC**.

16. L'appui sur la touche de fonction **F3** fait apparaître à l'écran le masque des constantes d'étalonnage. Chaque thermosalinographe Sea-Bird possède ses propres coefficients d'étalonnage qui doivent être recalculés tous les ans lors de la calibration de l'appareil. Une fois saisis, ces coefficients sont stockés dans le fichier constant .cfg par l'appui sur **F2**.

```

Orston Nouméa - B.Buisson - J.Grelet - 04.0 Nov 91

  ———— SONDE SBE Constantes d'étalonnage ————

    Température                               Conductivité
    N série : 338                               N série : 603
    A : 3.67582852e-03                          A : 2.17302275e-04
    B : 5.78941892e-04                          B : 4.04687796e-01
    C : 7.79066810e-06                          C : -4.04288962e0
    D : -2.04107153e-6                          D : 1.75111913e-04
    F0 : 2297.33                                M : 3.3

PgUp : Précédent , PgDn : Suivant , CtPgUp : Premier , CtPgDn : Dernier
F3  : Insertion , F4  : Ajout , F5  : Suppression 1 / 1
↓   : Champ Suivant , ↑   : Champ Précédent
F2  : Écrit. & Sortie , F9 : Écriture Disque , ESC : Sortie
  
```

Fig 7. Masque de saisie des coefficients d'étalonnage.

Il est conseillé, après saisie des coefficients, de faire une copie du fichier constant .cfg dans un fichier xxx.cfg où xxx représente le numéro de série de l'appareil. Cette méthode permet d'interchanger facilement les appareils sans avoir à re-saisir les coefficients.

17. L'appui sur la touche de fonction **F4** fait apparaître à l'écran le masque de configuration du logiciel. La touche **F1** permet de choisir, en ouvrant une nouvelle fenêtre, l'appareil de navigation.

```

Orston Nouméa - B.Buisson - J.Grelet - 04.0 Nov 91

  ———— Thermosalinographe SBE21 Configuration logiciel ————

    Systeme de navigation(F1): NMEA
    Emetteur ARGOS 0/N       : N
    Stockage sur disque 0/N  : 0
    Intervalle (sec)        : 300
    Nom du fichier          : b:voyager
    Multiplexeur ( COM1 )   : 0

    NMEA
    AUX
    MANNO
    MANUEL

PgUp : Précédent , PgDn : Suivant , CtPgUp : Premier , CtPgDn : Dernier
F3  : Insertion , F4  : Ajout , F5  : Suppression 1 / 6
↓   : Champ Suivant , ↑   : Champ Précédent
F2  : Écrit. & Sortie , F9 : Écriture Disque , ESC : Sortie
  
```

Fig 8. Masque de saisie des paramètres de configuration du logiciel.

Il est nécessaire d'entrer le chemin d'accès complet avec le nom du fichier (ex: b:voyager).

18. L'accès au paramètre de communication de chacun des appareils se fait en appuyant sur la touche **PageBas** autant de fois qu'il y a d'appareils (quatre ou cinq suivant les versions).

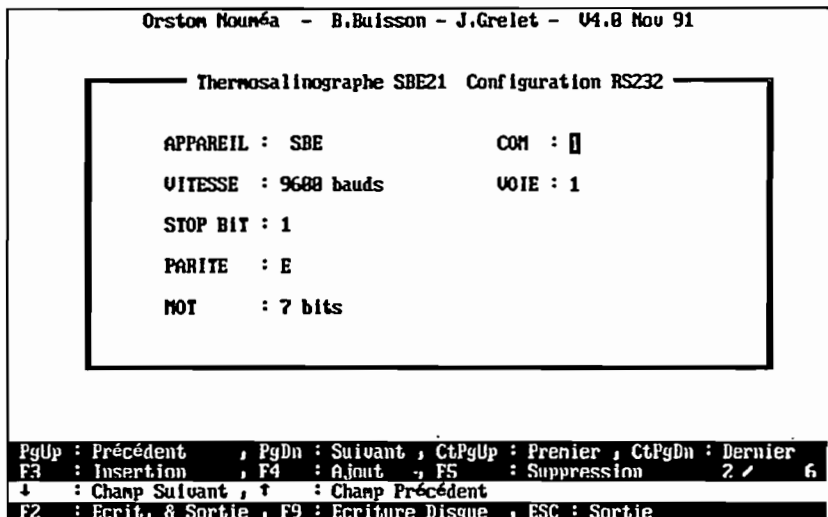


Fig 9. Masque de saisie des paramètres de communication d'une liaison série.

Les masques peuvent être saisis dans n'importe quel ordre, il suffit d'associer à l'appareil choisi (SBE, MVX, NALNO, NMEA ou ARGOS) les paramètres de communication correspondants.

Affichage du mode trace option (-t)

Le mode trace permet d'afficher à la place du masque **Dernière position du navire** une fenêtre où les caractères envoyés par l'appareil choisi vont s'afficher. Ce mode est utile lorsqu'il est nécessaire de vérifier l'établissement du dialogue entre l'appareil et le micro-ordinateur, lors de l'installation de l'équipement par exemple. Un seul appareil peut être 'tracé' en même temps.

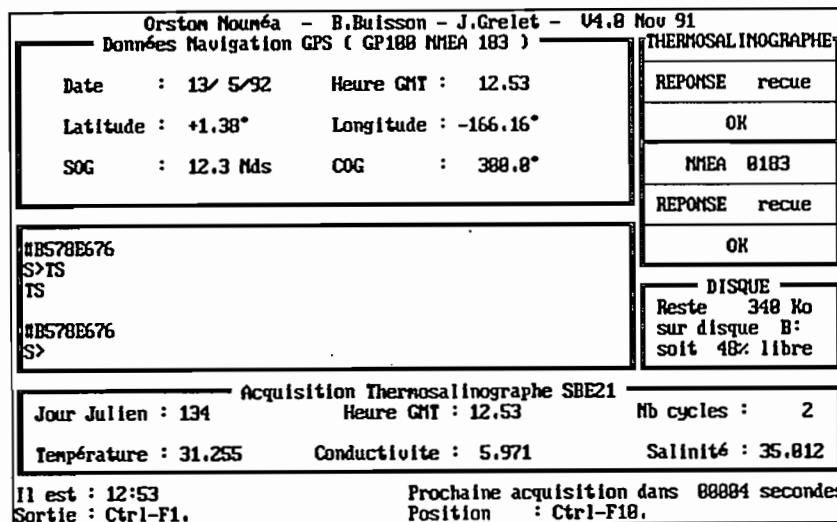


Fig 10. Exécution du programme avec choix du thermosalinographe en mode trace.

L'utilisateur doit entrer la commande suivante : thermo -tt.

| Orston Nouméa - B.Buisson - J.Grelet - V4.0 Nov 91 | | |
|--|----------------------|---|
| Mauvais GPS Veuillez entrer la position manuelle Dernière position correcte : Heure GMT: 12.54 Latitude : +1.38° Longitude : -166.16° | | THERMOSALINOGRAPHE |
| | | DEMANDE en cours |
| | | NMEA 0183 |
| | | ERREUR |
| \$GPGLL,3850.00N,09500.00,W*7E \$GPXTE,A,A,00.76,R,N*41 \$GPBWC,225310,3910.00,N,0955.00,W,014.5,I,,M,003.0,N,2*0E \$GPRMC,125456,A,0138.23,N,16616.34,W,12.3,300,138592,,E*34 \$GPRMB,A,00.77,R,1,2,3910.00N,09500.00,W,003.0,0 | | DISQUE |
| | | Reste 340 Ko sur disque B: soit 48% libre |
| Acquisition Thermosalinographe SBE21 | | |
| Jour Julien : 134 | Heure GMT : 12.54 | Nb cycles : 2 |
| Température : 31.255 | Conductivité : 5.971 | Salinité : 35.012 |
| Il est : 12:54:56 | | Prochaine acquisition dans 00013 secondes |
| Sortie : Ctrl-F1. | | Position : Ctrl-F10. |

Fig 11 Exécution du programme avec choix de l'appareil de navigation en mode trace.
L'utilisateur doit entrer la commande suivante : thermo -tn.

Utilisation du mode simulation option (-s)

Le mode simulation permet d'utiliser le programme sans que les appareils soient connectés au micro-ordinateur. Ce mode permet d'évaluer le logiciel ou de l'utiliser à des fins pédagogiques. Cette possibilité de simuler les appareils est un atout précieux pour la mise au point de nouvelles versions du logiciel. Pour simuler le thermosalinographe et l'appareil de navigation, entrer la commande suivante: thermo -snt

Les différentes options peuvent être utilisées simultanément. Exemple:

```
thermo -m -tt -snt
```

Cette commande appelle le menu au lancement du programme, affiche le mode trace pour le thermosalinographe et simule les deux appareils. Les options peuvent être entrées dans n'importe quel ordre.

7.6. Les fichiers présents sur la disquette programme

Les micro-ordinateurs utilisés sur les bateaux marchands sont des systèmes possédant uniquement deux lecteurs de disquettes. La disquette programme doit donc contenir le système d'exploitation ainsi que les programmes permettant de configurer correctement le système.

Fichiers présents:

COMMAND.COM: Interpréteur de commandes du système d'exploitation MS DOS.

AUTOEXEC.BAT: Fichier batch exécuté automatiquement lors de la procédure de chargement du DOS. Il doit contenir les instructions suivantes dans le cas de l'utilisation d'un portable Zenith:

```
mode el 10
keybfr
rtclock
thermo
```

MODE.COM: Programme permettant selon la syntaxe utilisée, de paramétrer le fonctionnement du micro-ordinateur. La commande `mode el 10` présente dans le fichier `autoexec.bat` permet d'éteindre automatiquement l'éclairage arrière de l'écran LCD au bout de 10 minutes. Il suffit d'appuyer sur une touche pour rallumer l'écran (**SHIFT** par exemple).

RTCLOCK.COM: Programme de gestion de l'heure système. Pour changer l'heure et la date du Zenith, entrer la commande suivante:

```
RTCLOCK MM-DD-YY HH:MM
```

THERMO.EXE: Programme d'acquisition en français, version 4.0 du thermosalinographe SBE21.

***.ECR:** Descripteurs d'écrans. Fichiers ASCII utilisés par le programme.

***.DSK:** Descripteurs de disques. Fichiers ASCII utilisés par le programme.

CONSTANT.CFG: Fichier ASCII contenant les coefficients de calibration du thermosalinographe utilisé par le programme.

CONFIG.CFG: Fichier ASCII contenant la configuration du programme.

8. RESULTATS

Nous avons réalisé avec ce système, depuis Novembre 1990, des enregistrements le long de plusieurs rails de navigation dans le Pacifique Ouest sur des bateaux marchands et lors de campagnes océanographiques.

Au total, depuis un an et demi, trois navires de commerce et deux navires océanographiques ont permis de réaliser 15 voyages représentant 88325 observations en général espacées de 2 à 5 minutes. Sur des navires de commerce filant 15-20 noeuds cela représente une observation environ tous les 1.2 à 1.5 miles. La résolution spatiale est bien supérieure à celle obtenue habituellement avec des seaux météorologiques (90 à 120 miles).

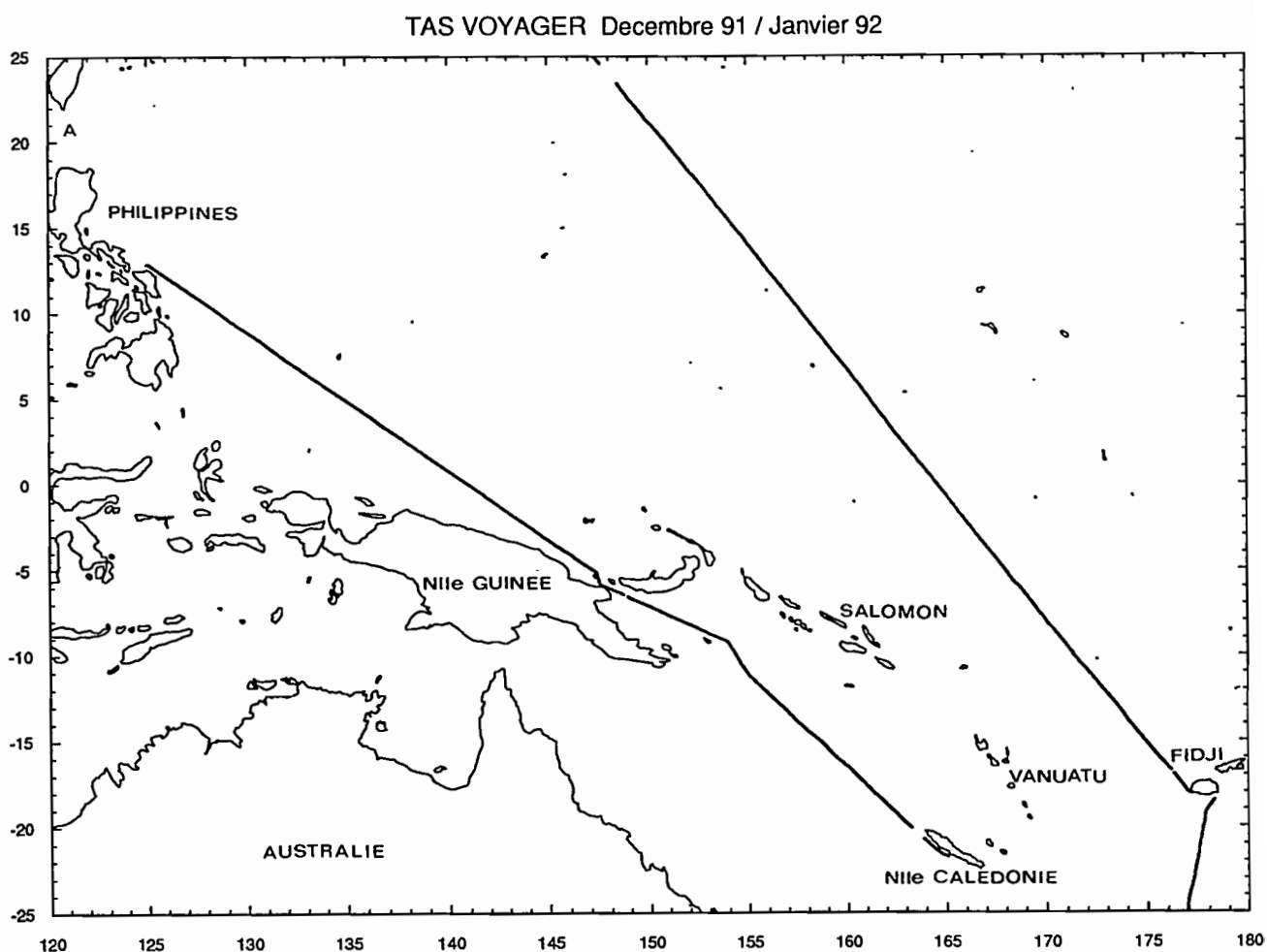


Fig 12. Distribution spatiale des mesures réalisées sur le T.A.S VOYAGER du 13 décembre 1991 au 27 janvier 1992 le long des trajets Nouvelle Calédonie/Philippines et Japon/Fidji.

TAS VOYAGER: aller / retour

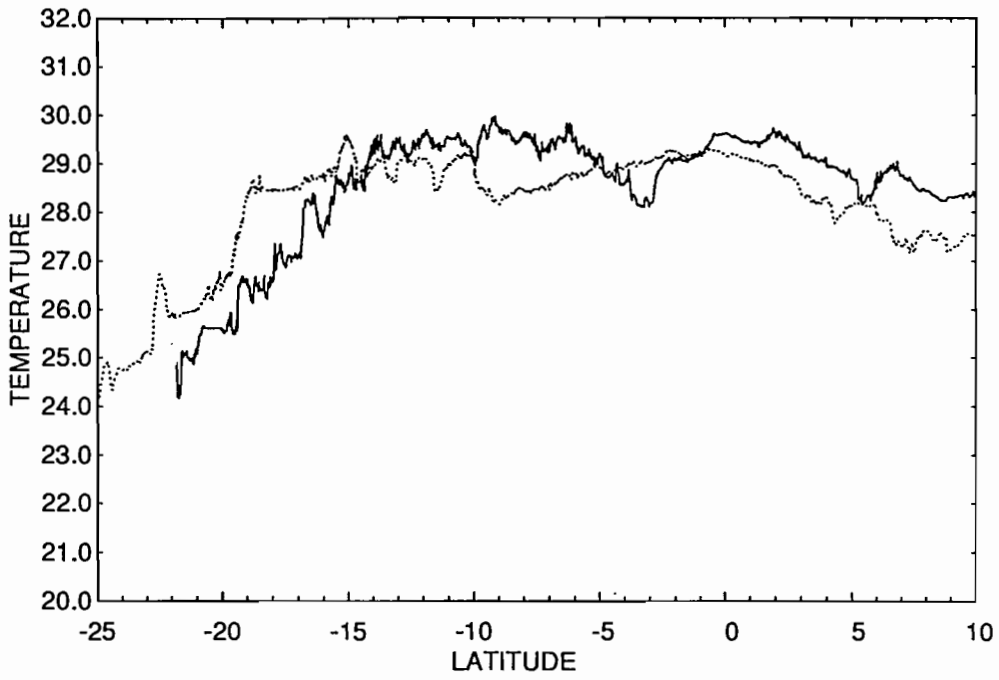


Fig13. Température de surface entre Nouvelle Calédonie/Phillippines (gras) et Japon/Fidji (pointillés)

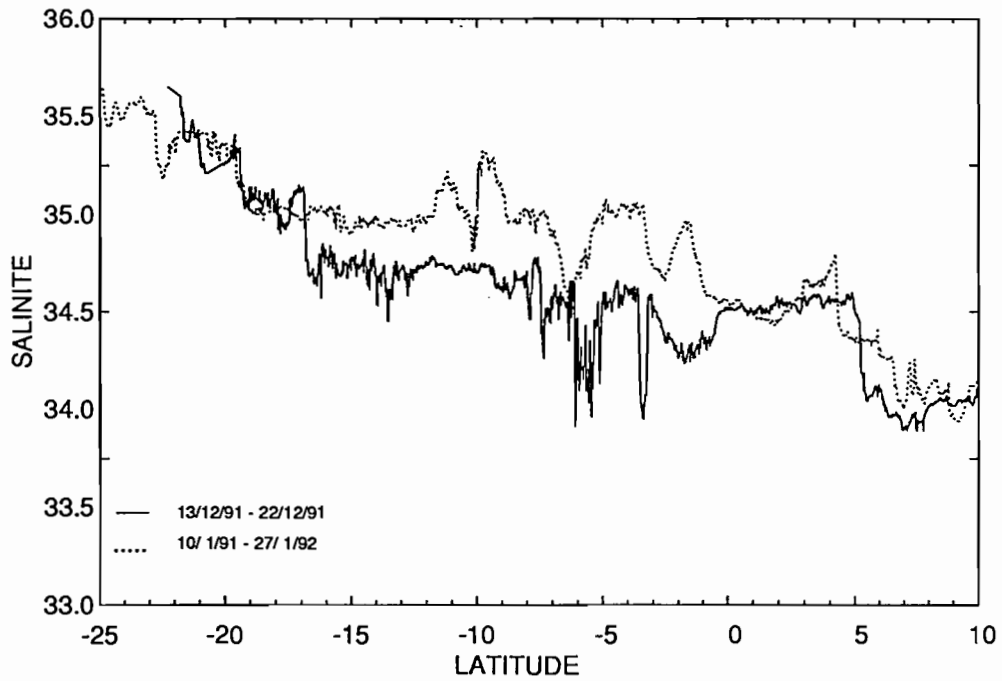


Fig14. Salinité de surface entre Nouvelle Calédonie/Phillippines (gras) et Japon/Fidji (pointillés).

Les premiers résultats, (Henin et Grelet, 1992), font apparaître que cette nouvelle technique automatique améliore très sensiblement la précision de la mesure de la salinité de surface mais aussi celle de la température de surface. Des inter-calibrations ont été effectuées lors des campagnes SURTROPAC avec une sonde CTD Sea-Bird modèle SBE09. Les mesures au thermomètre seau font apparaître un écart ($T_{CTD} - T_{seau}$) de $+0.60^{\circ}\text{C}$ (SURTROPAC15) à -0.13°C (COARE156-2) alors que l'écart pour le thermosalinographe ($T_{CTD} - T_{SBE21}$) est de -0.20°C (SURTROPAC15) à -0.30°C (COARE156-2). De même la dispersion des mesures diffère, l'écart type des différences étant bien plus faible pour le thermosalinographe (0.05 à 0.12°C) que pour le seau (0.34 à 0.48°C). L'écart de température observé s'explique très largement par le réchauffement de l'eau de mer lors de son parcours dans les conduites depuis la vanne de coque jusqu'au thermosalinographe.

L'amélioration est également sensible pour la mesure de la salinité car l'écart ($S_{CTD} - S_{SBE21}$) est relativement faible et constant pour le thermosalinographe (compris entre 0.000 et $+0.025$ psu), alors que pour le seau ($S_{CTD} - S_{seau}$) il varie de -0.130 et $+0.666$ psu d'une campagne à l'autre. Là encore, la dispersion des mesures est bien plus faible avec le thermosalinographe (0.011 à 0.024 psu) qu'avec le seau (0.097 à 0.192 psu).

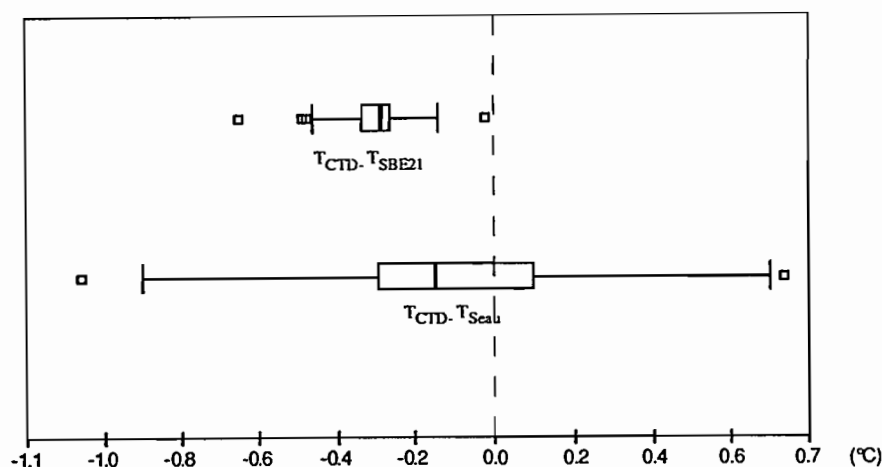


Fig 15. Boîte à moustache: comparaisons des écarts $T_{CTD} - T_{seau}$ et $T_{CTD} - T_{SBE21}$ calculés lors de la campagne COARE156-2.. La boîte contient 50% des données, les moustaches sont égales à 1.5 fois la largeur de la boîte.

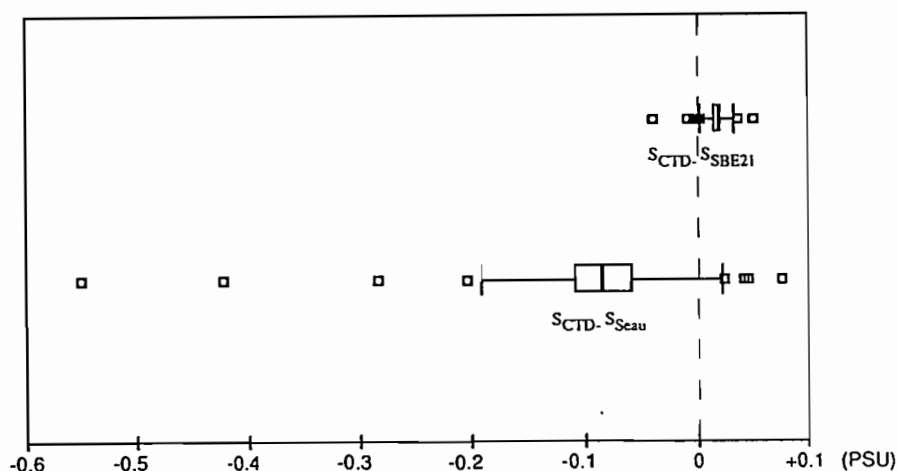


Fig 14. Boîte à moustaches: comparaisons des écarts $S_{CTD} - S_{seau}$ et $S_{CTD} - S_{SBE21}$ calculés lors de la campagne COARE156-2.

9. CONCLUSION

Au cours de la campagne COARE156-2 en février-mars 1992 (165°E-156°E, 20°S-8°N), il a été expérimenté une méthode d'échantillonnage à plus haute densité avec des mesures toutes les 15 secondes. Des traitements statistiques (moyenne, médiane et écart type), réalisés toutes les 300 secondes sur 24 données ont permis de mettre en évidence la bonne représentativité de la médiane par rapport à la moyenne. Cette technique permet de retirer les pics que l'on obtient avec une mesure instantanée tout en éliminant les mauvaises valeurs dues parfois à des erreurs de transmission. On peut considérer que la précision est améliorée par un facteur 5. Cette amélioration, utilisée en routine sur le N/O Le Noroît, sera installée progressivement sur les bateaux marchands.

Une autre amélioration prévue consiste à stocker les données dans un format binaire. Cette technique, outre l'intérêt de réduire la taille des fichiers de données, permettra à l'utilisateur, au moyen d'un programme de transcodage, d'obtenir des données ASCII au format et à la précision qu'il désire. Après utilisation de ce logiciel sur différents bateaux et par différentes équipes de recherche, il est en effet apparu nécessaire de pouvoir adapter le format des données aux besoins spécifiques des utilisateurs.

Toutes ces améliorations devraient être intégrées dans le logiciel au début du deuxième semestre 1992. Il est à noter que le programme peut très bien expédier les données en temps réel par ARGOS. La connexion avec un émetteur ne pose aucun problème et a déjà été réalisée au laboratoire. Il appartient aux organismes intéressés par le traitement en temps réel de développer une chaîne de traitement spécifique.

Un développement du logiciel a été réalisé lors des deux dernières campagnes, SURTROPAC16 et COARE156-2, pour effectuer des mesures de précipitations avec un capteur de pluie optique (STI modèle ORG-100), par l'intermédiaire d'une carte d'acquisition Analogique Numérique 12 bits de marque Metrabyte. Une version autonome de cette carte, pilotée par un micro-processeur est en cours de développement au laboratoire. Ce système sera interrogeable à distance via une liaison série, comme les autres appareils gérés par le programme, ce qui pourrait permettre son installation éventuelle sur des navires de commerce. Ce système permettra en outre l'acquisition de différents paramètres météorologiques (vent, humidité relative, température, pression atmosphérique, etc) et donc une meilleure compréhension des interactions océan-atmosphère dans les zones tropicales.

10. BIBLIOGRAPHIE

- Borland . **Turbo C Version 2.0.**
Manuel de l'utilisateur, 1988
- Buisson B. **SF Logiciel de saisie de fichiers ASCII.**
Manuel de référence. Laboratoire d'informatique,
ORSTOM centre de nouméa, 1989.
- Duncan Ray. **MSDOS avancé.** Le guide Microsoft du
programmeur en Assembleur et en Langage C.
Microsoft press, éditions P.S.I, Paris, 1989.
- Grelet J, Buisson B. **Mesure de température et commande
de processus pour analyseur de CO₂.** Notes techniques,
sciences de la mer, océanographie physique, N°6.
ORSTOM Centre de nouméa, 1992.
- Henin C, Grelet J. **Automatisation de la mesure de la
température et de la salinité de surface aux stations
cotières et sur les navires du réseau d'observation dans
l'océan Pacifique Tropical.** (Convention CORDET).
Rapport scientifique et technique, océanographie phy-
sique, ORSTOM Centre de Nouméa, 1992.
- Kernighan, Ritchie. **Le langage C.**
Traduit par Buffenoir, 7^e tirage. Masson, 1989.
- Lilen H. **8088-8086 Circuits périphériques.** Le contrôleur
programmable d'interruption PIC 8259. Le timer pro-
grammable 8253. Edition Radio, 1987.
- Magnavox. **Installation manual MX 1102/1107 GPS,**
1987.
- Magnavox. **Navigator manual MX 1102/1107 GPS.**
Chapitre 4. Interface ports. Pages 4-1 à 4-12 et R-8,
1987.
- Murray Sargent III, Richard L. Shoemaker. **The IBM-PC
from the inside out.** Addison Wesley Publishing
Company, 1986.
- Pronav. **GPS 100, personal navigator, owner's manual.**
1990.
- Sea-bird, **Seacat Thermosalinograph. Operating Manual**
Sea-Bird Electronics, Inc. Bellevue, Washington, 1990.
- Vasseur Daniel. **RT/CMOS sur AT et PS/2** Extrait de 'La
revue de l'utilisateur de l'IBM-PC' Février 89. Pages 20
à 36, 1989.

11. ANNEXE A: CARACTERISTIQUES DES APPAREILS

11.1. Thermosalinographe SBE21

11.1.1. Description technique

MECANIQUES

| | |
|------------|---------------------------|
| Boîtier: | PVC |
| Dimension: | 480 mm * 480 mm * 230 mm. |
| Poids: | 10 kgs. |

ALIMENTATION

| | |
|---------------|--|
| Tension: | Boite de jonction: 220 Volts. Thermosalinographe: 12 Volts. |
| Consommation: | 80 mA sous 10-15 Volts continu |

INTERFACES

| | |
|--------------|---|
| Format RS232 | 9600 Bauds, 7 bits, 1 bit stop, parité. impair |
|--------------|---|

PERFORMANCES

| | |
|----------------------------|---|
| Vitesse d'échantillonnage: | de 5 à 65535 secondes . |
| Gamme de mesure: | Température: -5 °C à 35 °C. Conductivité: 0 à 6.5 S/m/mois. |
| Précision: | Température: 0.01 °C/6 mois Conductivité: 0.001 S/m/mois. |
| Résolution: | Température: 0.001 °C Conductivité: 0.0001 S/m |
| Mémoire: | Mémoire statique CMOS de 64Koctets avec batterie de sauvegarde pour 2 ans. |

11.1.2. Commandes

Les commandes suivantes peuvent être exécutées après connexion du thermosalinographe à la boîte de jonction et celle-ci à un port série de l'ordinateur. Lancez le programme THERMO et entrez dans le mode terminal (cf chapitre 7: utilisation du programme THERMO). Appuyez sur la touche **Return** jusqu'à obtenir la réponse **S>** de l'appareil. En cas de non réponse, vérifiez les connexions et la configuration du port série utilisé. Les commandes décrites doivent être validées par l'appui sur la touche **Return**.

| | |
|-----------|--|
| DS | Commande Display Status: affiche à l'écran les informations d'état contenues dans la RAM |
| TS | Commande Take Sample: déclenche une mesure, envoyée à l'ordinateur après huit secondes. |

Ces deux commandes sont les seules qu'il est nécessaire de connaître pour vérifier le bon fonctionnement de l'appareil. Le lecteur trouvera une description complète des commandes dans le manuel d'utilisation "SEACAT THERMOSALINOGRAPH OPERATING MANUAL" fournis avec l'appareil (Sea-Bird, 1990).

11.2. Magnavox MX-1107

11.2.1. Description technique

MECANIQUES

| | |
|------------|---|
| Boîtier: | Etanche à l'aspersion, protégé contre la corrosion. |
| Dimension: | 432 mm * 419 mm * 356 mm. |
| Poids: | 38.5 kgs. |

ALIMENTATION

| | |
|---------------|--|
| Tension: | 100/115/222/230 V secteur 50 ou 60 Hz. |
| Consommation: | 175 W. |

INTERFACES

| | |
|--------------|--|
| Format RS232 | 4800 bauds, 7 bits de données, 1 bit de stop, parité impaire |
|--------------|--|

PERFORMANCES

| | |
|--------------------|-----------|
| Récepteur Transit: | 37 m RMS |
| Récepteur GPS: | 17 m RMS. |

11.2.2. Commandes

- Trames reconnues par le programme:

| | |
|---------------------------------------|-----|
| Passage satellite TRANSIT: | : M |
| Données satellite TRANSIT: | : X |
| Complément données satellite TRANSIT: | : Q |
| Données estime TRANSIT: | : O |
| Statut couverture GPS: | : 2 |
| Données GPS: | : 1 |

- Mode de transmission des trames:

| | |
|-----|---|
| : M | : spontanée, dès le passage satellite |
| : X | : spontanée, après calcul (8 à 10mn après: M) |
| : Q | : en réponse à la demande :K Q |
| : O | : en réponse à la demande :K O |
| : 2 | : en réponse à la demande :K 2 |
| : 1 | : en réponse à la demande :K 1 |

Le lecteur se reportera à la documentation du Magnavox pour avoir le détail des formats de chaque trame (Chapitre 4: *Interface ports*, pages 4-1 à 4-12).

11.3. Pronav GPS-100

11.3.1. Description technique

MECANIQUES

| | |
|------------|---|
| Boîtier: | Etanche à l'aspersion, protégé contre la corrosion. |
| Dimension: | 159 mm * 100 mm * 51 mm. |
| Poids: | 0.8 kgs. |

ALIMENTATION

| | |
|---------------|---|
| Tension: | Accus rechargeables. 230 V alternatif avec chargeur d'accus. |
| Consommation: | 245 mA en mode normal sur accus. 145 mA sur alimentation extérieure. |

INTERFACES

| | |
|----------|--|
| NMEA0183 | 4800 bauds, 8 bits, 1 bit stop, sans parité Trame BWC, GLL RMB, RMC R00, WPL, XTE. |
|----------|--|

PERFORMANCES

| | |
|-------------------------|---|
| Récepteur: | Acquisition de 8 satellites. |
| Temps d'acquisition: | 2 minutes en mode 2 D. 2,5 minutes en mode 3 D. 20 secondes en mode QuickFix. |
| Cadence de mise à jour: | une fois par seconde, en continu. |
| Précision: | Position: 15 mètres RMS Vitesse: 0.1 noeuds RMS. |
| Dynamique de poursuite: | 695 noeuds en vitesse, 3 g en accélération. |

11.3.2. Commandes

- Fréquence d'émission:

Sept trames de données envoyées spontanément toutes les deux secondes. Les trames envoyées par l'appareil sont les suivantes:

- GPRMC: Données GPS.
- GPRMB: Informations de navigation.
- GPR00: Identification des points de route entrés par l'utilisateur.
- GPWPL: Localisation d'un point de route.
- GPGLL: Position géographique de l'appareil.
- GPXTE: Erreur de position par rapport à la route.
- GPBWC: Cap et distance de l'appareil par rapport à un point de route.

- Format de la trame GPRMC:

GPRMC est la seule trame parmi les sept qui sont envoyées qui nous intéresse. En voici le format:

```
$GPRMC,HHMMSS,A,DDMM.MM,N,DDDMM.MM,W,XXX.X,XXX.X,DDMMYY,XXX.X,E*XX<CR><LF>
```

1 2 3 4 5 6 7 8 9 10 11 12

où

- 1 : Heure GMT
- 2 : A = Position valide, V = Perte couverture GPS
- 3 : Latitude (Degrés minutes)
- 4 : N = Nord, S = Sud
- 5 : Longitude (Degrés minutes)
- 6 : E = Est, W = Ouest
- 7 : Vitesse fond (Noeuds)
- 8 : Route fond (degrés)
- 9 : Date (Jour Mois Année)
- 10 : Variation magnétique
- 11 : Direction variation (E = Est, W = Ouest)
- 12 : Checksum

Le calcul de la checksum est réalisé par une opération logique (OU exclusif) sur les huit bits de tous les caractères constituant la trame, incluant les délimiteurs ',' compris entre, mais sans les inclure, les délimiteurs '\$' et '*'. Le résultat est converti en hexadécimal, 2 caractères ASCII (0-9, A-F) pour la transmission et le caractère de poids le plus fort est transmis en premier.

11.4. Centrale NALNO

11.4.1. Description technique

INTERFACES

Format RS232

Message de navigation type ADOP:

4800 bauds, 8 bits de données, 2 bit de stop, sans parité

Message de navigation pour l'ADCP (trame GPGGA):

2400 bauds, 8 bits de données, 2 bit de stop, sans parité

Message de navigation brut GPS MLR:

4800 bauds, 8 bits de données, 2 bit de stop, sans parité

Message de navigation intégrée NALNO (demande ORSTOM trame GPRMC):

4800 bauds, 8 bits de données, 2 bit de stop, sans parité

11.4.2. Commandes

- Fréquence d'émission:

Une trame de données envoyée spontanément toutes les six secondes

- Format de la trame GPRMC:

\$GPRMC, HHMMSS, DDMM.MM, N, DDDMM.MM, W, XXX.X, XXX.X, DDMMYY, XXX.X, E<CR><LF>

1 2 3 4 5 6 7 8 9 10

où

- 1 : Heure GMT
- 2 : Latitude (Degrés minutes)
- 3 : N = Nord, S = Sud
- 4 : Longitude (Degrés minutes)
- 5 : E = Est, W = Ouest
- 6 : Vitesse fond (Noeuds)
- 7 : Route fond (degrés)
- 8 : Date (Jour Mois Année)
- 9 : Variation magnétique
- 10 : Direction variation (E = Est, W = Ouest)

- Format de la trame GPGGA:

\$GPGGA, HHMMSS, DDMM.MM, N, DDDMM.MM, W, 1, 0, 01, 0, M, 0, M<CR><LF>

1 2 3 4 5

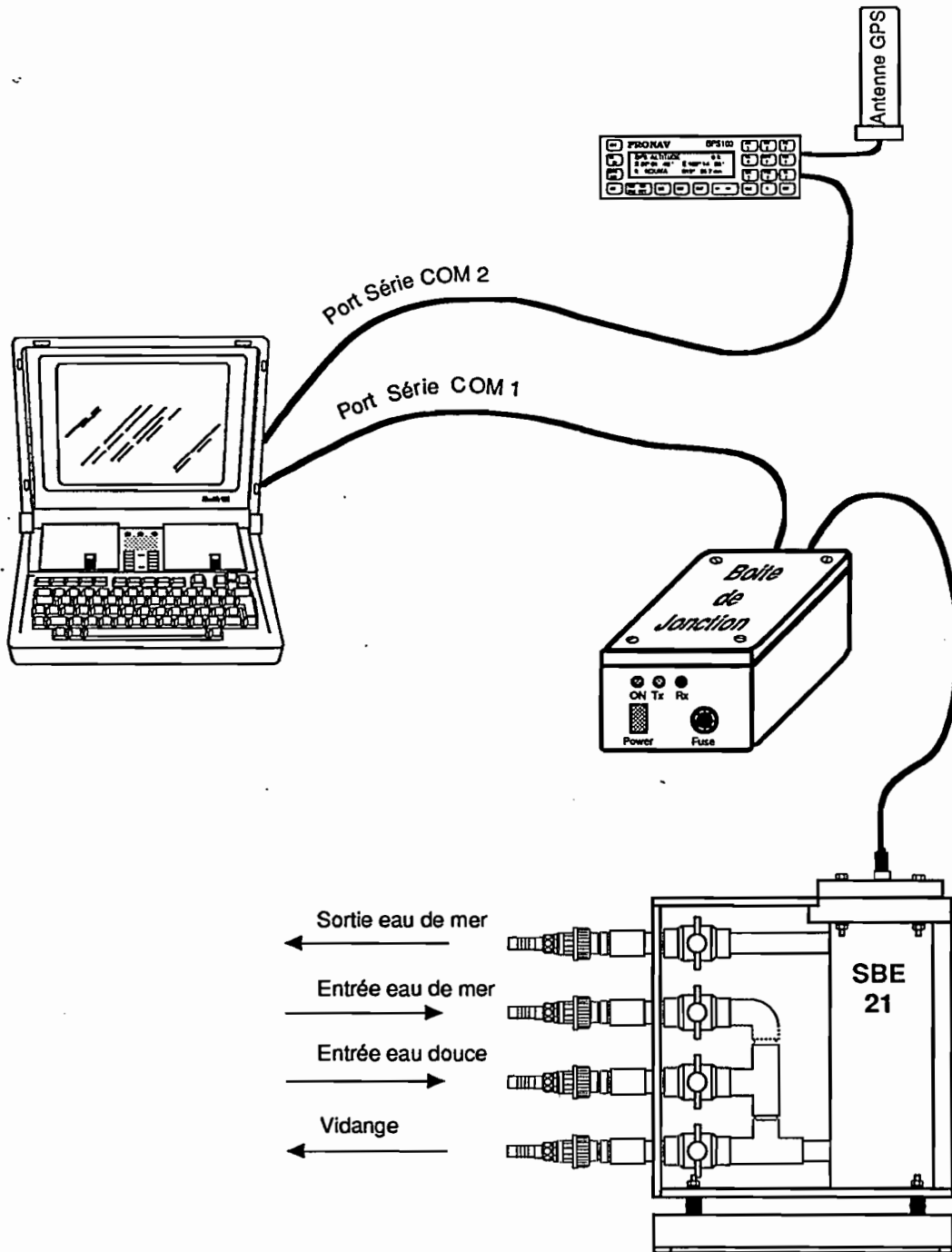
où

- 1 : Heure GMT
- 2 : Latitude (Degrés minutes)
- 3 : N = Nord, S = Sud
- 4 : Longitude (Degrés minutes)
- 5 : E = Est, W = Ouest

Les autres champs sont des constantes sans signification.

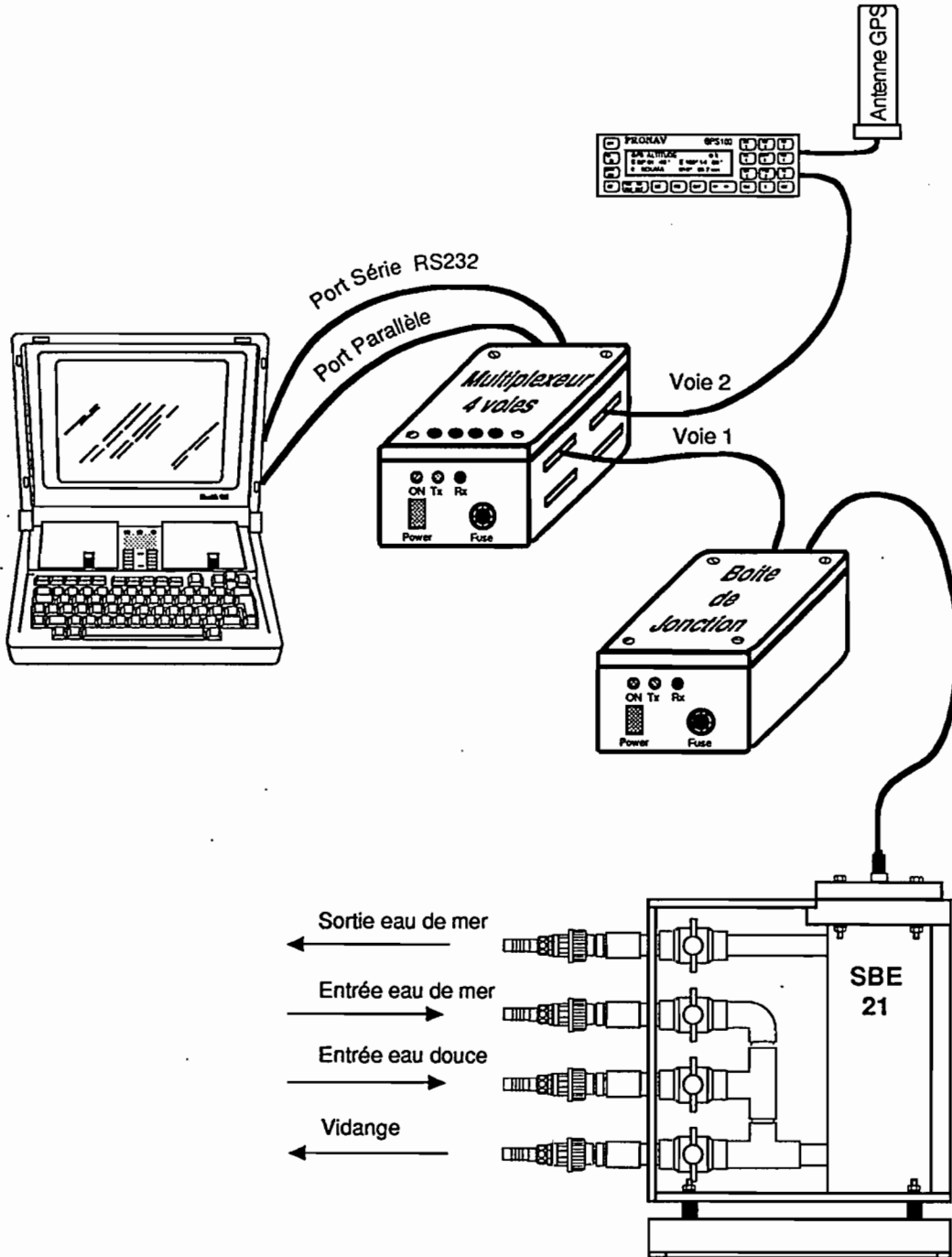
12. ANNEXE B: REALISATION TECHNIQUE

12.1. Synoptique du système d'acquisition



Thermosalinographe SBE21
Système d'acquisition
avec deux ports série

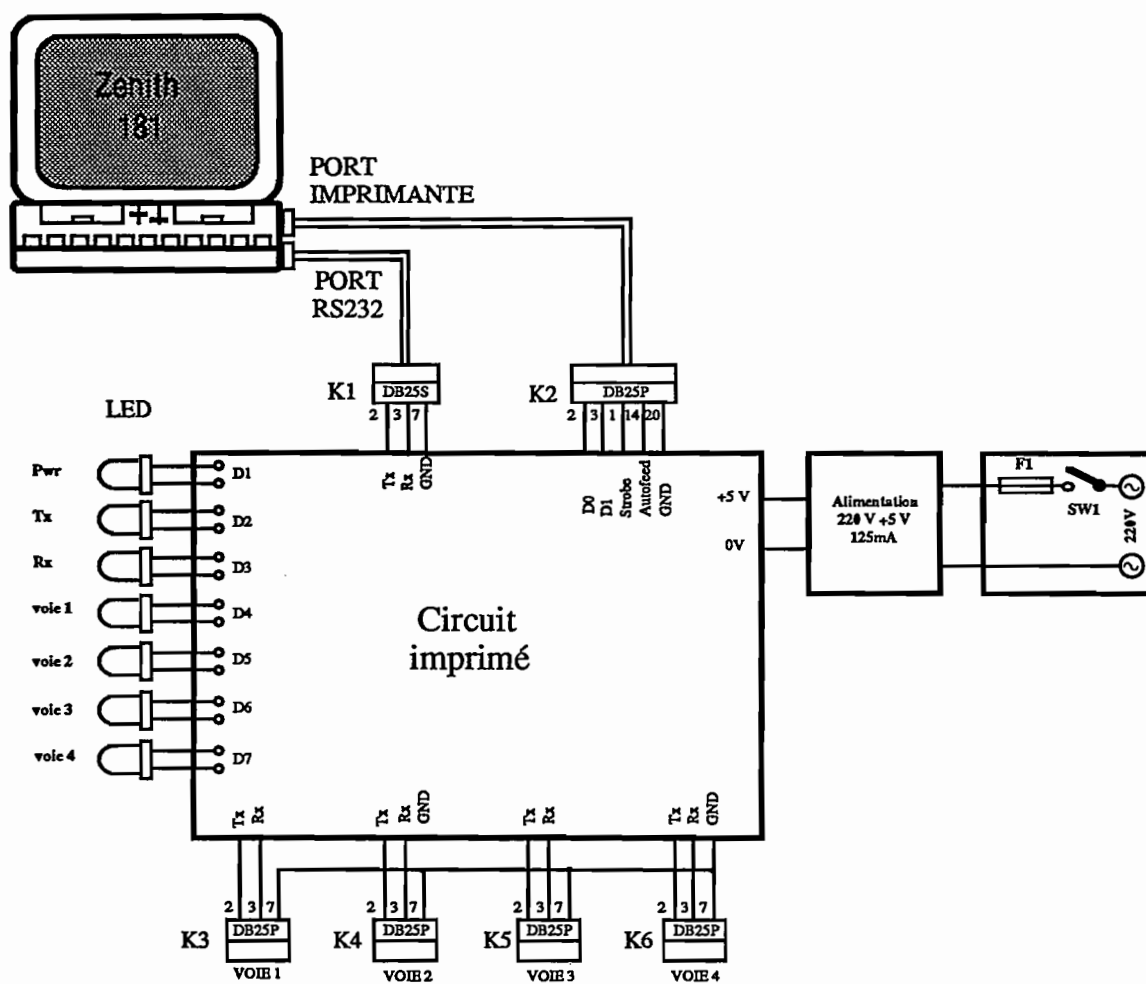
12.2. Synoptique du système d'acquisition avec multiplexeur



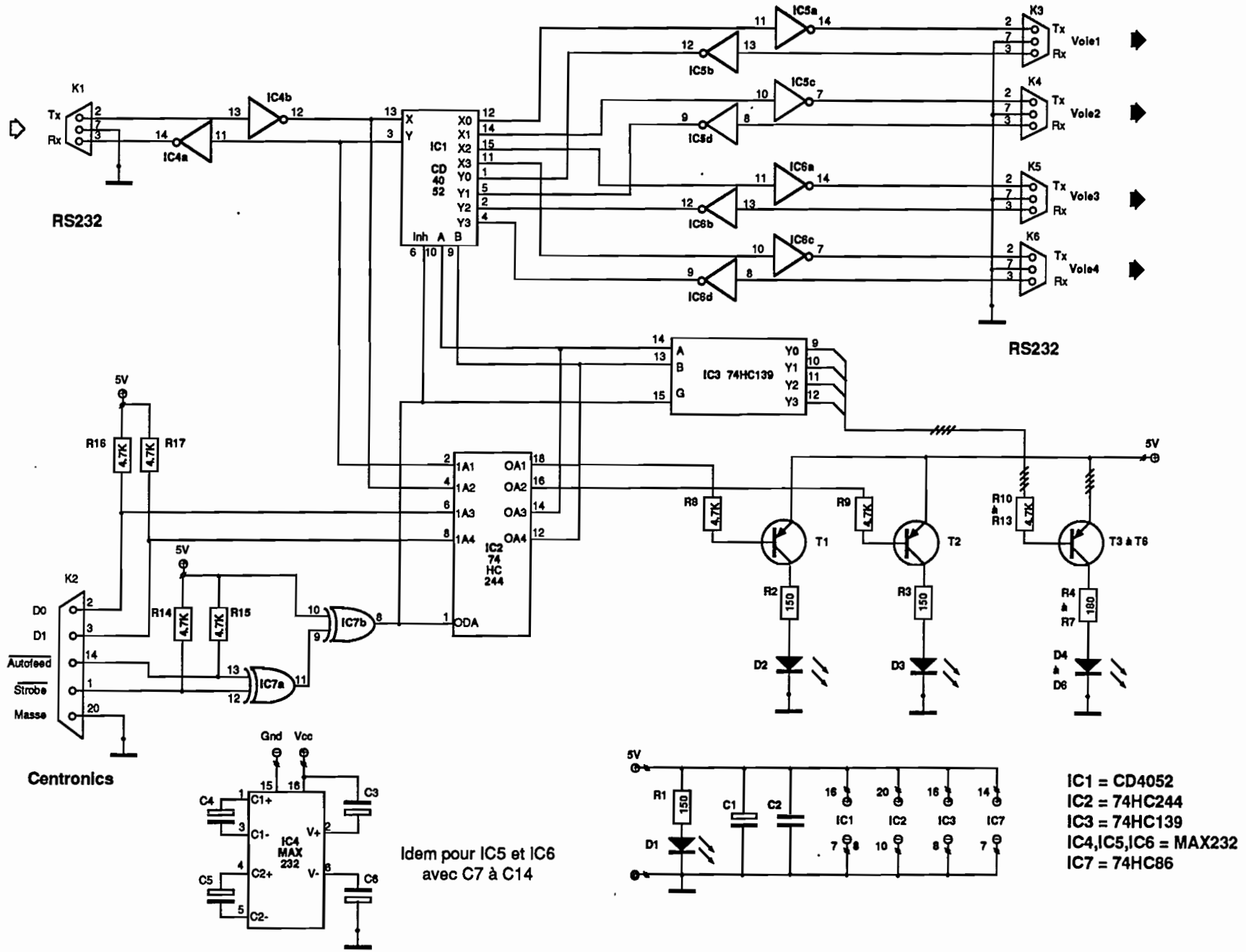
Thermosalinographe SBE21
Système d'acquisition avec un port série
et multiplexeur

12.3. Réalisation d'un multiplexeur de voies série

12.3.1. Câblage



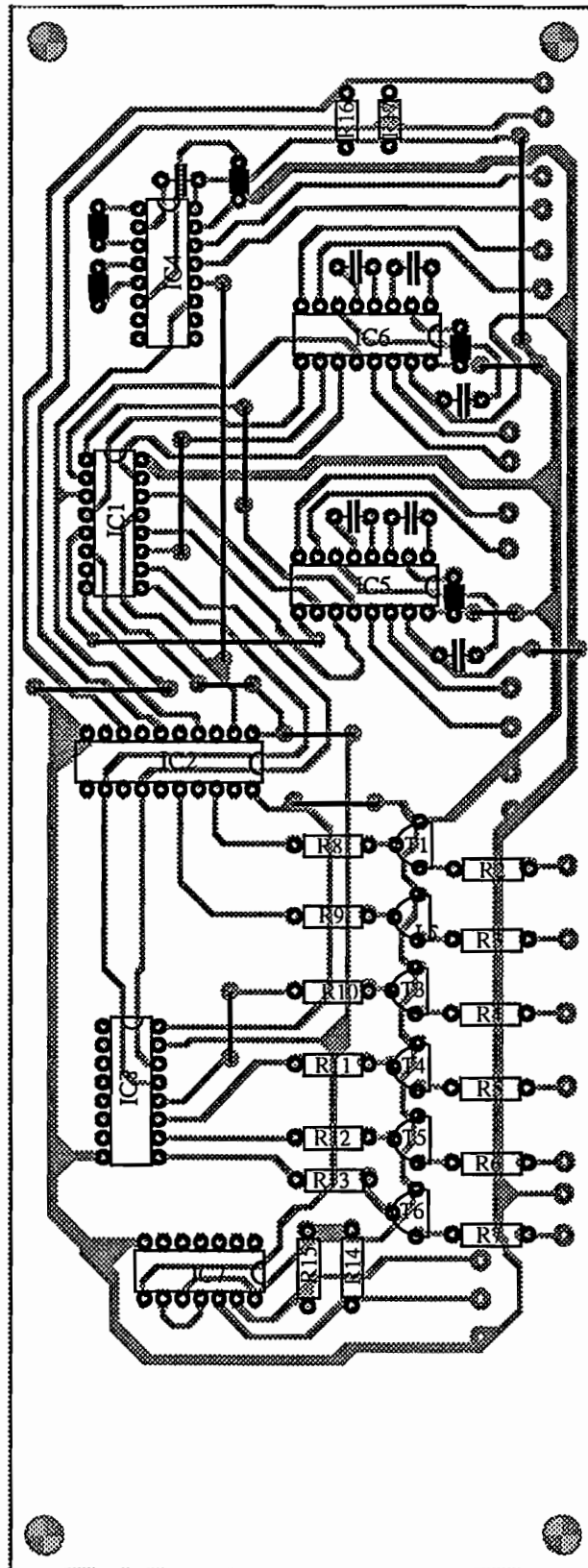
12.3.2. Schéma électronique



Idem pour IC5 et IC6 avec C7 à C14

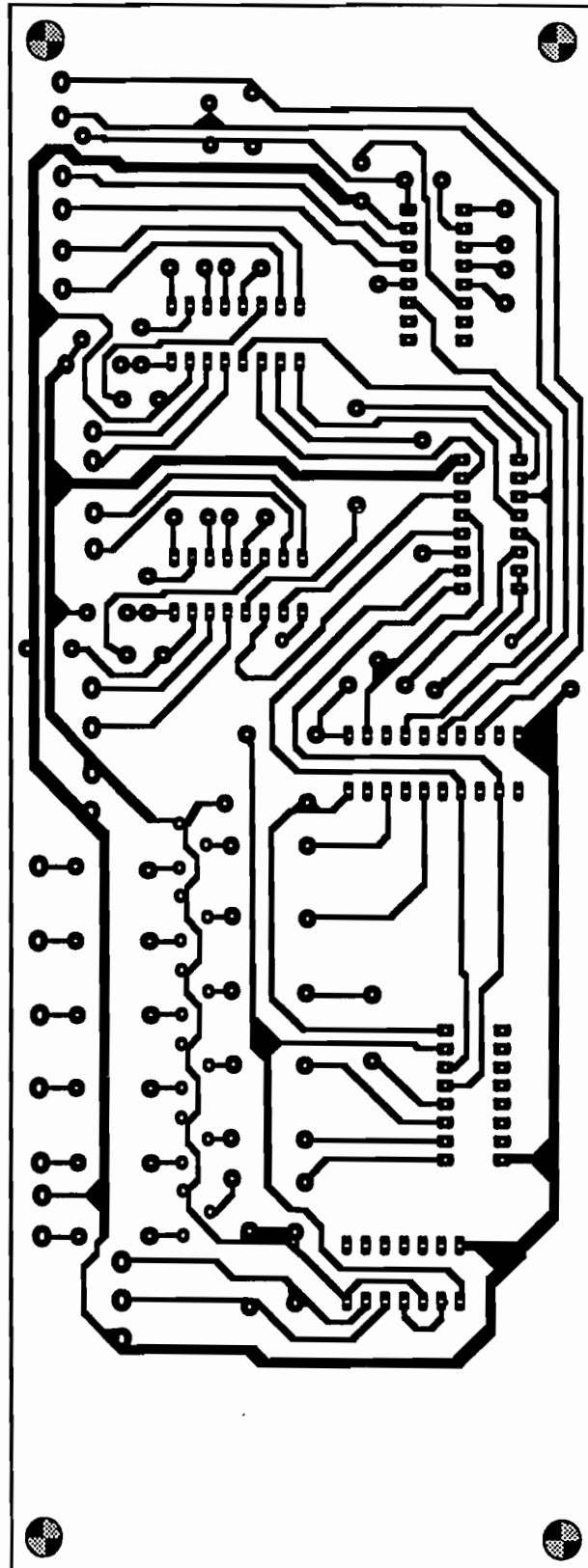
- IC1 = CD4052
- IC2 = 74HC244
- IC3 = 74HC139
- IC4, IC5, IC6 = MAX232
- IC7 = 74HC86

12.3.3. Implantation des composants



- D 1
- D 0
- Rx
- Tx K1
- Rx
- Tx K: 6
- Tx K 5
- Rx
- Tx K 4
- Tx K 3
- Rx
- +5V
- Masse
- D2
- D3
- D4
- D5
- D6
- Masse
- D7
- Autofeed
- Strobe
- Masse

12.3.4. Circuit imprimé



12.3.5. Nomenclature



MULTIPLEXEUR DE LIAISONS SERIE RS 232 LABORATOIRE SURTROPAC

| Références | Description | Quantité |
|------------|---|----------|
| R1 à R3 | Résistances 1/4 W 150 Ohms | 3 |
| R4 à R7 | Résistances 1/4 W 180 Ohms | 5 |
| R8 à R17 | Résistances 1/4 W 4700 Ohms | 10 |
| D1 | LED rouge 5mm | 1 |
| D2 | LED orange 5mm | 1 |
| D3 | LED verte 5mm | 1 |
| D4 à D6 | LED rouge 3mm | 4 |
| C1 | Capacité électrochimique 1000 μ F , 35V | 1 |
| C2 | Capacité 100 nF , 35V | 1 |
| C3 à C14 | Capacités électrochimique 10 μ F , 35V | 12 |
| T1 à T6 | Transistor PNP BC 307A | 6 |
| IC1 | Double multiplexeur 4 entrées CD4052 | 1 |
| IC2 | Octuple buffer 3 états 74HC244 | 1 |
| IC3 | Double décodeur/démultiplexeur 1 par 4 74HC139 | 1 |
| IC4 à IC6 | Double récepteur/transmetteur RS232 | 1 |
| IC7 | Quadruple OU Exclusif 2 entrées 74HC86 | 1 |
| K1 | Connecteur 25 broches femelle + capot DB25S | 1 |
| K2 | Connecteur 25 broches mâle + capot DB25P | 1 |
| K3 à K6 | Connecteur 25 broches femelle DB25S | 4 |
| SW1 | Interrupteur bipolaire | 1 |
| | Module d'alimentation CA-CC 5V - 500 mA modèle PM534D | 1 |
| | Support de circuit intégré tulip 14 broches | 1 |
| | Support de circuit intégré tulip 16 broches | 5 |
| | Support de circuit intégré tulip 20 broches | 1 |
| | Embase CEE tripolaire | 1 |
| | Cordon d'alimentation CEE 2 pôles + terre | 1 |
| | Câble téléphone 3 paires (m) | 2 |
| | Picots à souder | 20 |

12.4. Préparation d'un thermosalinographe

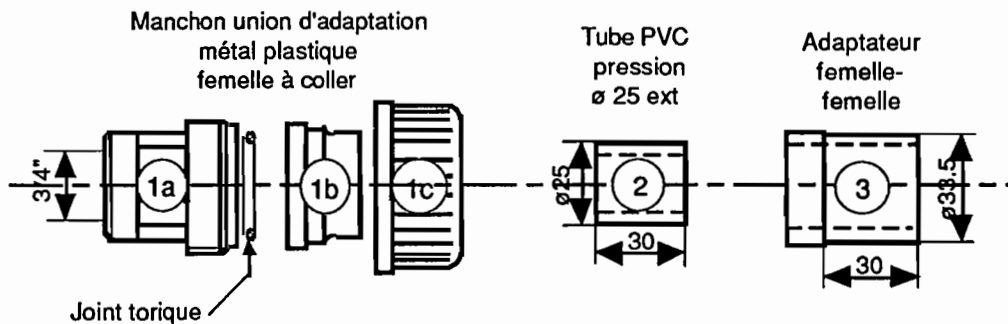
Il est nécessaire avant toute installation sur un navire, de réaliser quelques travaux qui permettront de préparer correctement le thermosalinographe et d'éviter ainsi, toute surprise de dernière minute. Le lecteur trouvera tous les plans cotés et schémas dans les pages suivantes.

12.4.1. Support de fixation

Il est conseillé de réaliser un support en acier galvanisé sur lequel on fixera le thermosalinographe avec quatre amortisseurs de vibrations. Ce support de coût réduit présente l'avantage de pouvoir être rapidement soudé sur une structure du bateau. Les dimensions de ce support sont de 480 mm * 245 mm et il est réalisé avec de la cornière en L de 50 mm. La protection contre tout risque de corrosion est assurée par une application de deux couches de peinture PA10 de chez Epiglass. Le support et le bâti en PVC du thermosalinographe sont percés de trous de Ø8 mm aux quatre coins d'un rectangle de 430 * 200 mm pour la fixation des amortisseurs.

12.4.2. Assemblage des accouplements

Le thermosalinographe est livré avec quatre sorties PVC au diamètre américain. Il est nécessaire de fabriquer des adaptateurs afin de pouvoir utiliser le matériel acheté aux normes Européennes. Ces adaptateurs (n°3 sur le schéma) sont réalisés dans un manchon Nicholl de diamètre intérieur 25 mm. Le diamètre extérieur de ce manchon doit être rectifié sur un tour à 33.5 mm et sur une longueur de 30 mm. Les quatre pièces ainsi réalisées sont collées et assemblées dans les manchons femelles d'origine.



Remarque:

Toutes les pièces devant être assemblées par collage doivent être soigneusement dégraissées et dépolies au papier de verre. Lorsque les pièces ont été enduites de colle, elles doivent être assemblées rapidement sans mouvement de torsion. Laisser ensuite sécher 24 heures.

La partie n°1b du manchon union est ensuite raccordée par collage à l'adaptateur n°3 par l'intermédiaire du tube PVC pression n°2. Ne pas oublier d'intercaler l'écrou n°1c avant l'assemblage.

12.4.3. Check-list d'installation



CHECK-LIST

Installation Thermosalinographe SBE21

LABORATOIRE SURTROPAC

SYSTEME À PRÉPARER

- Thermosalinographe avec support et boîte de jonction.....
- Câble + connecteur 4 broches RMG 4 FS.....
- Récepteur Pronav GPS100 avec alimentation et connecteur RS232.....
- Antenne extérieure et mat aluminium.....
- Presse étoupe Ø10 et connecteur BNC à souder.....
- Micro-ordinateur Zenith avec alimentation secteur.....
- Disquette système MSDOS.....
- Disquette programme..... Version..... Nb.....
- Disquette data..... Nb.....
- Documentation thermosalinographe..... Version.....
- Documentation GPS..... Version.....

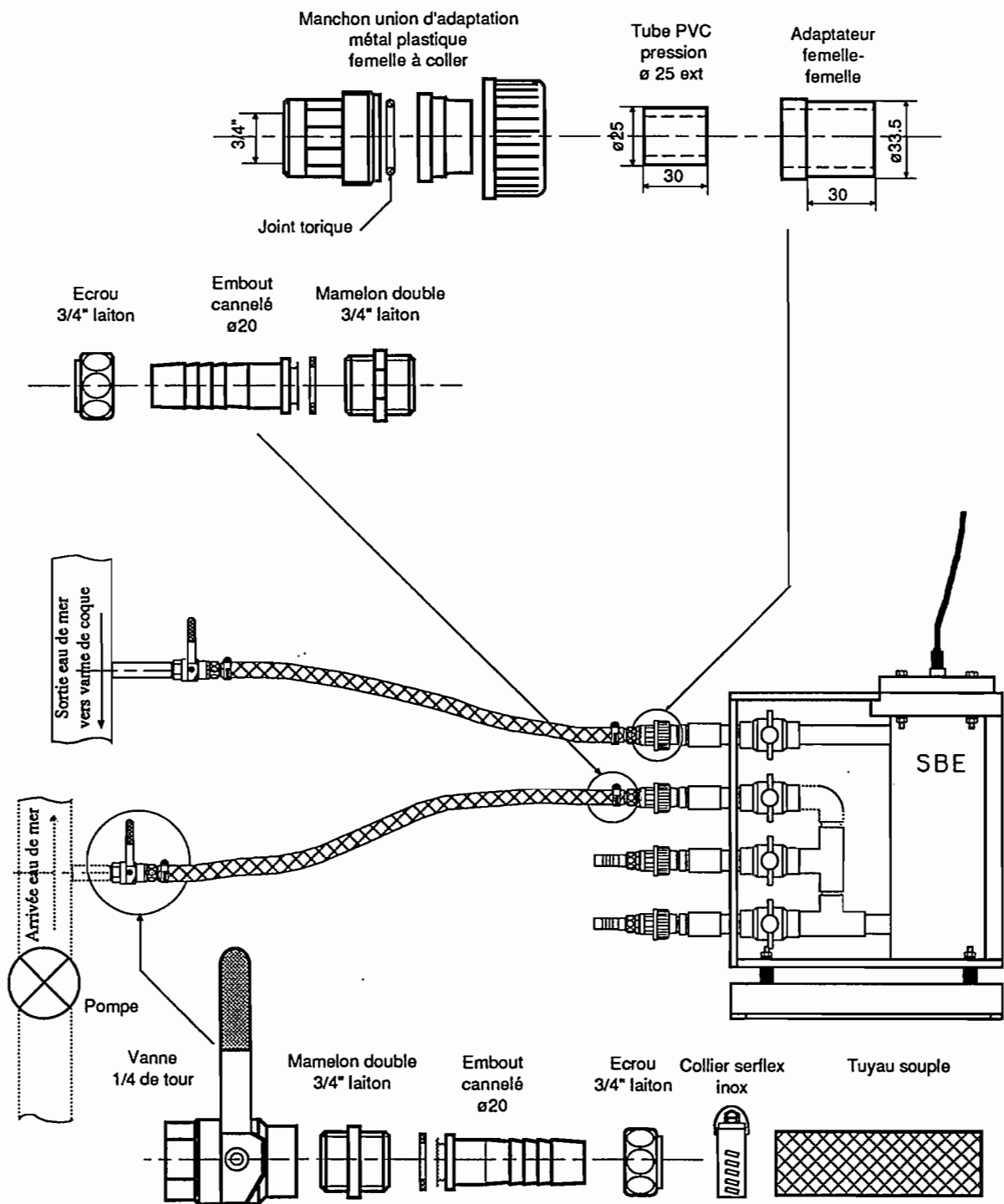
MATÉRIEL À PRÉPARER

- Vanne 3/4 pouce..... Nb.....
- Manchon union d'adaptation 3/4 pouce femelle en PVC..... Nb.....
- mamelon double 3/4 pouce en laiton..... Nb.....
- Embout cannelé Ø20 en PVC avec écrou de fixation 3/4 pouce en laiton..... Nb.....
- Collier de serrage Ø30 en inox..... Nb.....
- Tuyau plastique renforcé Ø15..... Mètres.....
- Tuyau plastique Ø15..... Mètres.....
- Ruban téflon.....
- Pâte pour raccords filetés+ filasse.....
- Colle PVC.....
- Graisse silicone.....
- Scotch 33+, scotch 23 et scotchfil.....
- Bande velcro autocollante.....
- Câble téléphone 2 paires.....
- Boîtier d'alimentation 4 prises.....

OUTILS À PRÉPARER

- Caisse à outils électronique complète.....
- Fer à souder portable Weller (gaz) et soudure étain.....
- Perceuse et forets au nitrure de titane.....
- Pistolet à air chaud et gaine rétractable.....
- Clef à pipe et clef plate 7/16.....
- Clef à molette.....
- Pince multiprise.....
- Couteau, cutter.....

12.4.4. Schéma d'implantation à bord d'un navire



**Thermosalinographe SBE 21
Préparation des raccords en PVC**

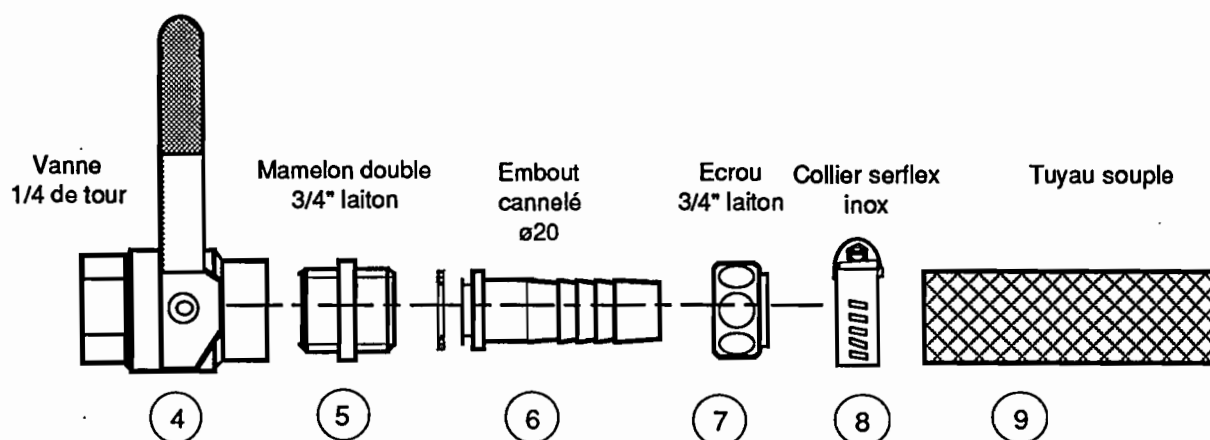
12.4.5. Installation sur site

Le choix de l'emplacement du thermosalinographe dépend en grande partie des possibilités d'alimentation en eau de mer dans la salle des machines. La prise d'eau doit se situer le plus près possible de la traverse alimentant le système de refroidissement du moteur afin d'éviter le réchauffement de l'eau par les tuyaux d'alimentation. L'idéal est de pouvoir se placer juste après les pompes du circuit de refroidissement.

La connexion sur ce circuit de refroidissement peut poser des problèmes lorsqu'aucune prise d'eau n'est disponible. Les conduites d'eau de mer sont souvent réalisées avec des aciers spéciaux (bronze d'aluminium) nécessitant, pour le branchement, un outillage spécialisé pas toujours disponible à bord. Une visite préalable sur le bateau est souvent nécessaire afin d'explorer calmement toutes les solutions possibles.

Il est très important de s'assurer du bon écoulement de l'eau dans l'appareil. La pression d'alimentation du circuit d'eau de mer en sortie des pompes est généralement de l'ordre de deux bars. Il est nécessaire de relier la sortie du thermosalinographe à une vanne de coque dont la profondeur n'excède pas huit mètres. Vérifier aussi que la pression d'alimentation en eau douce est supérieure à la pression présente sur la vanne de coque, sinon l'appareil ne pourra jamais être rincé.

Dans toutes les installations réalisées à ce jour, le branchement sur les canalisations a été fait soit directement par soudure d'un tube d'acier galvanisé, soit avec un mamelon double en laiton vissé dans un filetage existant. Le diamètre utilisé est toujours de $\frac{3}{4}$ pouce ce qui permet d'y fixer directement une vanne $\frac{1}{4}$ de tour en laiton chromé. Le raccordement entre la vanne (n°4) et le manchon union du thermosalinographe est réalisé avec du tuyau plastique renforcé (n°9) Tricoclar AL Ø15 intérieur emmanché à chaque extrémité sur un embout cannelé (n° 6) Ø20 mm et serré avec le collier serflex (n° 8). Ne pas oublier d'insérer l'écrou (n° 7) en laiton au préalable. Un mamelon double en laiton (n° 5) permet, sur le thermosalinographe et sur la vanne, de raccorder chaque extrémité du tuyau.



Toutes les pièces sont assemblées de préférence avec de la pâte à raccords filetés utilisée conjointement avec de la filasse. De cette façon on est pratiquement sûr de réaliser une étanchéité parfaite à chaque assemblage.

Le drain pourra se faire directement dans la cale du navire, cette sortie n'étant utilisée que pour vidanger l'appareil ou prélever un échantillon d'eau de mer.

La liaison entre le thermosalinographe et la passerelle est réalisée avec du câble quatre conducteurs blindé. Le passage du câble est la partie la plus longue et la plus pénible à réaliser et ne doit se faire qu'avec l'accord du chef mécanicien. Il est parfois possible d'utiliser des câbles en 'spares' entre la salle de contrôle de la machine et la passerelle. Après repérage des conducteurs, on peut souder le connecteur femelle étanche quatre broches du thermosalinographe (RMG 4FS) au câble dans la salle des machines et le connecteur mâle de la boîte de jonction (MS3106A-14S-2P) au câble à la passerelle.

Le micro-ordinateur, la boîte de jonction, le multiplexeur et le récepteur satellite sont fixés sur une planche en bois elle-même placée à la passerelle. L'antenne du récepteur est placée dans un endroit bien dégagé et le câble passé par un presse-étoupe.

12.4.6. Description de l'ensemble moteur/pompe

Certaines installations peuvent nécessiter la présence d'un système de pompage lorsqu'il n'est pas possible d'utiliser ou de se brancher sur le système de refroidissement du bord. Un ensemble moteur/pompe a été ainsi installé sur le N/O Alis de l'ORSTOM.

Nous avons porté notre choix sur un moteur 220 V triphasé accouplé à une pompe marine Johnson. Un équipement identique est déjà utilisé pour le refroidissement du groupe frigorifique de ce navire. La maintenance s'en trouve simplifiée car les pièces détachées sont communes aux deux systèmes.

Le matériel utilisé est le suivant:

- Pompe Johnson modèle F7B-3000 avec rotor en Néoprène, le débit est de 57 l à 1400 T/min sous 1.8 bar.
- Accouplement élastique Paulstra.
- Moteur triphasé 220 Volts Leroy Somer 1.1 Kw.
- Discontacteur triphasé + relais thermique Télémécanique.

L'ensemble est fixé sur un bâti en acier galvanisé. L'accouplement élastique doit être alésé aux dimensions des axes du moteur et de la pompe. Lors de la fabrication du support, ne pas oublier d'ovaliser les trous de fixation du moteur afin de pouvoir aligner le montage par la suite.

Le raccordement de la prise d'eau d'alimentation à la pompe et de la pompe au thermosalinographe sera réalisé de la même manière qu'au paragraphe précédent.

13. ANNEXE C

13.1. Structure du descripteur de fichier config.dsk

```
config;ASCII;

AUTO:0

2

config;7;|;
  navigateur;          CHAINE:7
  argos;               CHAINE:1
  sample_interval;    ENTIER:4
  stockage;           CHAINE:1
  nomfichier;         CHAINE:20
  multiplexeur;       CHAINE:1
  nbre_voies;         ENTIER:1

saisie( config )

rs232;7;|;
  appareil;           CHAINE:7
  rsport;             ENTIER:1
  voie;              ENTIER:1
  vitesse;           ENTIER:4
  bitstop;           ENTIER:1
  pair;              CHAINE:1
  longueur;          ENTIER:1

saisie( rs232 )

config( rs232 )6
```

13.2. Exemple de fichier config.cfg

```
MVX      |N| 60|N|b:voyager      |N|2
SBE      |1|1|9600|1|E|7
MVX      |2|2|4800|1|N|8
MVX_GPS  |2|2|4800|1|N|8
NALNO    |1|2|4800|1|N|8
NMEA     |1|2|4800|1|N|8
ARGOS    |1|2|2400|1|N|8
```

13.3. Structure du descripteur de fichier constant.dsk

```
constant;ASCII;

AUTO:0

1

constant;12;|;
  tserie;          ENTIER:3
  tempa;           CHAINE:14
  tempb;           CHAINE:14
  tempc;           CHAINE:14
  tempd;           CHAINE:14
  tempf0;          REEL : 7.2
  cserie;          ENTIER: 3
  cona;            CHAINE:14
  conb;            CHAINE:14
  conc;            CHAINE:14
  cond;            CHAINE:14
  conm;            REEL :3.1

saisie( constant )

constant
```

13.4. Exemple de fichier constant.cfg

```
603|3.67582852e-03|5.78941892e-04|7.79066810e-06|-2.04107153e-6|2297.33
603|2.17302275e-04|4.84687796e-01|-4.04208962e0 |1.75111913e-04|3.3
```

13.5. Format du fichier de données

| | | | | | | | | | | | | | | | | | | | |
|-------------------|---|-----|-------|--------|--------|--------|---------|--------|--------|--|--|--|--|--|--|--|--|--|--|
| Début acquisition | 0 | JJJ | HH.MM | YY | | | | | | | | | | | | | | | |
| Estime TRANSIT | 1 | JJJ | HH.MM | jjj | hh.mmm | sDD.MM | sDDD.MM | TT.TTT | SS.SSS | | | | | | | | | | |
| Acquisition GPS | 2 | JJJ | HH.MM | jjj | hh.mmm | sDD.MM | sDDD.MM | TT.TTT | SS.SSS | | | | | | | | | | |
| Position manuelle | 3 | JJJ | HH.MM | jjj | hh.mmm | sDD.MM | sDDD.MM | | | | | | | | | | | | |
| THERMO seul | 4 | JJJ | HH.MM | TT.TTT | SS.SSS | | | | | | | | | | | | | | |
| Fix TRANSIT | 5 | JJJ | HH.MM | jjj | hh.mmm | S11.11 | SLLL.LL | | | | | | | | | | | | |
| Fin acquisition | 6 | JJJ | HH.MM | YY | | | | | | | | | | | | | | | |

où

JJJ et HH.MM représentent l'heure du PC et
jjj et hh.mmm représentent l'heure satellite (ou manuelle).

s est le signe de la latitude et de la longitude.

TT.TTT est la température en degrés celcius,
SS.SSS est la salinité en PSU.

13.6. Calendrier des jours juliens

(Année normale: 365 jours)

| Jour | Jan | Fev | Mar | Avr | Mai | Juin | Jul | Aout | Sep | Oct | Nov | Dec | Jour |
|------|-----|-----|-----|-----|-----|------|-----|------|-----|-----|-----|-----|------|
| 1 | 001 | 032 | 060 | 091 | 121 | 152 | 182 | 213 | 244 | 274 | 305 | 335 | 1 |
| 2 | 002 | 033 | 061 | 092 | 122 | 153 | 183 | 214 | 245 | 275 | 306 | 336 | 2 |
| 3 | 003 | 034 | 062 | 093 | 123 | 154 | 184 | 215 | 246 | 276 | 307 | 337 | 3 |
| 4 | 004 | 035 | 063 | 094 | 124 | 155 | 185 | 216 | 247 | 277 | 308 | 338 | 4 |
| 5 | 005 | 036 | 064 | 095 | 125 | 156 | 186 | 217 | 248 | 278 | 309 | 339 | 5 |
| 6 | 006 | 037 | 065 | 096 | 126 | 157 | 187 | 218 | 249 | 279 | 310 | 340 | 6 |
| 7 | 007 | 038 | 066 | 097 | 127 | 158 | 188 | 219 | 250 | 280 | 311 | 341 | 7 |
| 8 | 008 | 039 | 067 | 098 | 128 | 159 | 189 | 220 | 251 | 281 | 312 | 342 | 8 |
| 9 | 009 | 040 | 068 | 099 | 129 | 160 | 190 | 221 | 252 | 282 | 313 | 343 | 9 |
| 10 | 010 | 041 | 069 | 100 | 130 | 161 | 191 | 222 | 253 | 283 | 314 | 344 | 10 |
| 11 | 011 | 042 | 070 | 101 | 131 | 162 | 192 | 223 | 254 | 284 | 315 | 345 | 11 |
| 12 | 012 | 043 | 071 | 102 | 132 | 163 | 193 | 224 | 255 | 285 | 316 | 346 | 12 |
| 13 | 013 | 044 | 072 | 103 | 133 | 164 | 194 | 225 | 256 | 286 | 317 | 347 | 13 |
| 14 | 014 | 045 | 073 | 104 | 134 | 165 | 195 | 226 | 257 | 287 | 318 | 348 | 14 |
| 15 | 015 | 046 | 074 | 105 | 135 | 166 | 196 | 227 | 258 | 288 | 319 | 349 | 15 |
| 16 | 016 | 047 | 075 | 106 | 136 | 167 | 197 | 228 | 259 | 289 | 320 | 350 | 16 |
| 17 | 017 | 048 | 076 | 107 | 137 | 168 | 198 | 229 | 260 | 290 | 321 | 351 | 17 |
| 18 | 018 | 049 | 077 | 108 | 138 | 169 | 199 | 230 | 261 | 291 | 322 | 352 | 18 |
| 19 | 019 | 050 | 078 | 109 | 139 | 170 | 200 | 231 | 262 | 292 | 323 | 353 | 19 |
| 20 | 020 | 051 | 079 | 110 | 140 | 171 | 201 | 232 | 263 | 293 | 324 | 354 | 20 |
| 21 | 021 | 052 | 080 | 111 | 141 | 172 | 202 | 233 | 264 | 294 | 325 | 355 | 21 |
| 22 | 022 | 053 | 081 | 112 | 142 | 173 | 203 | 234 | 265 | 295 | 326 | 356 | 22 |
| 23 | 023 | 054 | 082 | 113 | 143 | 174 | 204 | 235 | 266 | 296 | 327 | 357 | 23 |
| 24 | 024 | 055 | 083 | 114 | 144 | 175 | 205 | 236 | 267 | 297 | 328 | 358 | 24 |
| 25 | 025 | 056 | 084 | 115 | 145 | 176 | 206 | 237 | 268 | 298 | 329 | 359 | 25 |
| 26 | 026 | 057 | 085 | 116 | 146 | 177 | 207 | 238 | 269 | 299 | 330 | 360 | 26 |
| 27 | 027 | 058 | 086 | 117 | 147 | 178 | 208 | 239 | 270 | 300 | 331 | 361 | 27 |
| 28 | 028 | 059 | 087 | 118 | 148 | 179 | 209 | 240 | 271 | 301 | 332 | 362 | 28 |
| 29 | 029 | | 088 | 119 | 149 | 180 | 210 | 241 | 272 | 302 | 333 | 363 | 29 |
| 30 | 030 | | 089 | 120 | 150 | 181 | 211 | 242 | 273 | 303 | 334 | 364 | 30 |
| 31 | 031 | | 090 | | 151 | | 212 | 243 | | 304 | | 365 | 31 |

CALENDRIER DES JOURS JULIENS

(Année bissextile: 366 jours)

| Jour | Jan | Fev | Mar | Avr | Mai | Juin | Jul | Aout | Sep | Oct | Nov | Dec | Jour |
|------|-----|-----|-----|-----|-----|------|-----|------|-----|-----|-----|-----|------|
| 1 | 001 | 032 | 061 | 092 | 122 | 153 | 183 | 214 | 245 | 275 | 306 | 336 | 1 |
| 2 | 002 | 033 | 062 | 093 | 123 | 154 | 184 | 215 | 246 | 276 | 307 | 337 | 2 |
| 3 | 003 | 034 | 063 | 094 | 124 | 155 | 185 | 216 | 247 | 277 | 308 | 338 | 3 |
| 4 | 004 | 035 | 064 | 095 | 125 | 156 | 186 | 217 | 248 | 278 | 309 | 339 | 4 |
| 5 | 005 | 036 | 065 | 096 | 126 | 157 | 187 | 218 | 249 | 279 | 310 | 340 | 5 |
| 6 | 006 | 037 | 066 | 097 | 127 | 158 | 188 | 219 | 250 | 280 | 311 | 341 | 6 |
| 7 | 007 | 038 | 067 | 098 | 128 | 159 | 189 | 220 | 251 | 281 | 312 | 342 | 7 |
| 8 | 008 | 039 | 068 | 099 | 129 | 160 | 190 | 221 | 252 | 282 | 313 | 343 | 8 |
| 9 | 009 | 040 | 069 | 100 | 130 | 161 | 191 | 222 | 253 | 283 | 314 | 344 | 9 |
| 10 | 010 | 041 | 070 | 101 | 131 | 162 | 192 | 223 | 254 | 284 | 315 | 345 | 10 |
| 11 | 011 | 042 | 071 | 102 | 132 | 163 | 193 | 224 | 255 | 285 | 316 | 346 | 11 |
| 12 | 012 | 043 | 072 | 103 | 133 | 164 | 194 | 225 | 256 | 286 | 317 | 347 | 12 |
| 13 | 013 | 044 | 073 | 104 | 134 | 165 | 195 | 226 | 257 | 287 | 318 | 348 | 13 |
| 14 | 014 | 045 | 074 | 105 | 135 | 166 | 196 | 227 | 258 | 288 | 319 | 349 | 14 |
| 15 | 015 | 046 | 075 | 106 | 136 | 167 | 197 | 228 | 259 | 289 | 320 | 350 | 15 |
| 16 | 016 | 047 | 076 | 107 | 137 | 168 | 198 | 229 | 260 | 290 | 321 | 351 | 16 |
| 17 | 017 | 048 | 077 | 108 | 138 | 169 | 199 | 230 | 261 | 291 | 322 | 352 | 17 |
| 18 | 018 | 049 | 078 | 109 | 139 | 170 | 200 | 231 | 262 | 292 | 323 | 353 | 18 |
| 19 | 019 | 050 | 079 | 110 | 140 | 171 | 201 | 232 | 263 | 293 | 324 | 354 | 19 |
| 20 | 020 | 051 | 080 | 111 | 141 | 172 | 202 | 233 | 264 | 294 | 325 | 355 | 20 |
| 21 | 021 | 052 | 081 | 112 | 142 | 173 | 203 | 234 | 265 | 295 | 326 | 356 | 21 |
| 22 | 022 | 053 | 082 | 113 | 143 | 174 | 204 | 235 | 266 | 296 | 327 | 357 | 22 |
| 23 | 023 | 054 | 083 | 114 | 144 | 175 | 205 | 236 | 267 | 297 | 328 | 358 | 23 |
| 24 | 024 | 055 | 084 | 115 | 145 | 176 | 206 | 237 | 268 | 298 | 329 | 359 | 24 |
| 25 | 025 | 056 | 085 | 116 | 146 | 177 | 207 | 238 | 269 | 299 | 330 | 360 | 25 |
| 26 | 026 | 057 | 086 | 117 | 147 | 178 | 208 | 239 | 270 | 300 | 331 | 361 | 26 |
| 27 | 027 | 058 | 087 | 118 | 148 | 179 | 209 | 240 | 271 | 301 | 332 | 362 | 27 |
| 28 | 028 | 059 | 088 | 119 | 149 | 180 | 210 | 241 | 272 | 302 | 333 | 363 | 28 |
| 29 | 029 | 060 | 089 | 120 | 150 | 181 | 211 | 242 | 273 | 303 | 334 | 364 | 29 |
| 30 | 030 | | 090 | | 151 | | 212 | 243 | | 304 | | 365 | 30 |
| 31 | 031 | | 091 | | 152 | | 213 | 244 | | 305 | | 366 | 31 |

14. ANNEXE D

14.1. Coût du système

Cette liste de prix n'est pas exhaustive et est calculée sur des prix "arrivée à Nouméa". De plus, certains prix sont susceptibles de baisser rapidement ou certains composants de ne plus être disponibles sous la référence citée (micro-ordinateur, récepteur GPS etc...).

| | |
|---|-------------|
| Récepteur GPS Pronav GPS100 | 11725.00 |
| Kit antenne extérieure | 1420.00 |
| Pompe F7B-3000 rotor Néoprène | 2027.00 |
| Moteur électrique 110/220V | 2000.00 |
| Accouplement | 1000.00 |
| Contacteur de commande | 550.00 |
| Relais thermique | 200.00 |
| Support | 250.00 |
| Thermosalinographe Sea-Bird SBE21 | 43615.00 |
| 50 mètres de câble | 2200.00 |
| Antifouling | 440.00 |
| Vannes, accouplements PVC, raccords en bronze | |
| tuyaux PVC et plastique, colliers | 1000.00 |
| Support | 250.00 |
| Micro-ordinateur portable Zenith Supersport | 12000.00 |
| Multiplexeur 4 voies | 1500.00 |
| Total: | 80177.00 FF |

14.2. Adresses des fournisseurs

WAECO FRANCE

Pompe marine Johnson
Parc d'activité de la siagne n°21
06210 Mandelin
Fax: 93 47 81 66

JOHNSON PUMP AB

Pompe marine Johnson
PO Box 1436, Nastagan 19, S-701 14
OREBRO Sweden
Tél: 46 (19) 21 83 00
Fax: 46 (19) 13 48 54
Telex: 73559 johpumps

Compagnie Commerciale d'Electronique

Récepteur GPS
12 rue Charles Tellier
BP 153
13307Marseille cédex 14
Tél: 91 67 09 90
Fax: 91 02 93 46
Télex: 410987F

PRONAV

Récepteur GPS
11206 Thompson Avenue
Lenexa, KS 66219
1-800-800-1020
(913) 599-1515

SEA-BIRD

Thermosalinographe
1808-136 th Place NE
Bellevue, Washington 98005 USA
Tél: (206) 643 9866
Fax: (206) 643 9954
Telex: 292915 SBEI

15. GLOSSAIRE

| | |
|------------------|--|
| ARGOS | Localisation et collecte de données par satellite. |
| ASCII | American Standard Code for Information Interchange. |
| COARE | Coupled Ocean Atmosphere Response Experiment. |
| CORDET | Commission de Coordination de la Recherche dans les Départements et Territoires d'Outre Mer. |
| GPS | Global Positioning System (système de positionnement global). |
| IFREMER | Institut Français de Recherche pour l'Exploitation de la Mer. |
| NALNO | Centrale d'acquisition et de traitement des données de navigation. |
| ORG | Optical Rain Gauge (capteur optique de précipitations). |
| ORSTOM | Institut Français de Recherche Scientifique pour le Développement en Coopération. |
| PSU | Practical Salinity Unit (Unité de Salinité Pratique). |
| RAM | Random Access Memory (mémoire vive). |
| ROM | Read Only Memory. (mémoire morte). |
| SBE | Sea-Bird Electronic Inc. |
| SSS | Sea Surface Salinity (salinité de surface). |
| SST | Sea Surface Temperature (température de surface). |
| STI | Scientific Technologie Inc. |
| SURTROPAC | SURveillance TROPicale du PACifique. |
| TOGA | Tropical Ocean Global Atmosphere. |

16. INDEX

- accouplements 84
- ADCP 17
- application 27
- arme_delai 35
- ASCII 19
- AT 28
- a_rebours 35
- bâti 84
- bibliothèque de gestion d'écrans 46
- bibliothèque de gestion de fichiers 47
- bits 19
- boite de jonction 56
- capteur de conductivité 13
- capteur de température 13
- cellule de conductivité 14
- Centronics 20
- circuit de refroidissement 87
- circuit imprimé 82
- climat 9
- CMOS_D_T 37
- coefficients 13
- commande TS 11; 13
- commandes 73
- composants 81
- compteurs 34
- COMPUSAL 15
- config.cfg 50
- configuration 50
- connecteur étanche 88
- connexions 53
- constant.cfg 50
- contamination 14
- CORDET 9
- couches OSI 27
- coût du système 93
- dclose 49
- dcreate 48
- dedit 48
- delai_claque 36
- dérive 13; 15
- desarme_delai 35
- descripteur de fichier 47
- descripteurs d'écrans 45
- descripteurs de fichiers 45
- destructeur 40
- dget 49
- dispersion des mesures 67
- disquette 56
- dload 48
- dopen 48
- dpos 49
- dput 49
- DS 73
- dsave 48
- eau de mer normale 15
- écart type 67
- échelle de salinité pratique 14
- ECOQA 9
- électrodes 14
- enregistrement 45
- environnement multitâches 25
- étalonnage 13
- etat_delai 35
- Ethernet 27
- fenêtre edit 55
- fenêtre menu 55
- fenêtres d'affichage 56
- fichier config.cfg 89
- fichier config.dsk 89
- fichier constant.dsk 90
- fichier de données 90
- first_instance 49
- fixation 84
- fonctions d'interface 30
- fréquence 13
- gestionnaire d'interruption 29
- get_cmos_date_time 37
- get_pc_date_time 37
- GPGGA 76
- GPRMC 75
- GPS 16
- horloge temps réel 36
- IBM-PC 19
- init_delais 34
- init_trame 33
- INSTANCE 48
- inter-calibrations 67
- interruptions 25
- interruptions matérielle 28
- IRQ 28
- ISO 27
- JOHNSON 94
- jours juliens 91
- antenne du récepteur 88
- liaison série 15
- lib_paquet 33
- lib_pool 33
- lib_trame 33

liste des signaux 38
 maffiche 46
 Magnavox 74
 Magnavox 11
 Magnavox 16
 manchon 84
 masques 45
 masque de saisie 47
 masque des constantes d'étalonnage 60
 mclose 47
 menu 59
 menu principal 23
 méthodes 26
 mget 47
 microprocesseur 28
 mode simulation 44; 62
 mode terminal 23; 59
 mode trace 44; 61
 modulaire 26
 mopen 46
 mput 47
 mrefresh 46
 msaisit 46
 multiplexeur 12; 53
 multiplexeur de voies série 79
 multiplexeurs 20
 NALNO 11; 16; 76
 new_trame 33
 niveau TRAME 32
 niveaux d'interruption 28
 NMEA-0183 11
 nomenclature 83
 norme RS-232 19
 objet 26
 OMEGA 16
 options de la ligne de commande 59
 paquet_pret 33
 PIC 28
 polling 41; 42
 pompe 12; 88
 position du navire 58
 programmation par couches 27
 programmation par objets 26
 programme multitâches 25
 programme THERMO 23; 27
 Pronav 12; 16; 53; 56; 74; 94
 pr_serie 31
 PSU 14
 put_paquet 33
 PVC 84
 rapport de conductivité 14
 récepteurs GPS 11
 reset_paquet 33
 ressource logicielle 26
 ressources matérielles 26
 résultats 65
 RS232 12
 rsactive 32
 rsclose 31
 rsdefault 32
 rsdesactive 32
 rsgetc 31
 rslock 32
 rsoff 31
 rson 31
 rsopen 30
 rsputc 31
 rsputs 31
 rsunlock 32
 RT/CMOS 36
 Salinité de surface 66
 sauvegarde 57
 schéma du multiplexeur 80
 SCISOFT 11
 SCTERM 11
 SEA-BIRD 94
 SEASAVE 11
 SEASOFT 11
 seau météorologique 9
 self 40
 send_appli 33
 SERIE 30
 set_pc_date_time 37
 SF 45
 signaux 26; 38
 sonde CTID 11
 sonde Sea Cat 11
 SSS 9
 SST 9
 stockage 14
 strobe 20
 structure 26
 superviseur 25; 40
 superviseur de tâches 25
 SURTROPAC 9
 système d'acquisition 77
 système d'acquisition avec multiplexeur 78
 tâches 26; 27
 tâche de stockage 43
 tâches communicantes 37
 tampons circulaires 30
 température de surface 66
 term_delais 35
 thermosalinographe Sea-Bird 11
 this 40
 tinit 47
 TOGA 9

traite 41
TRAME 33
trame GPGGA 17
trame GPRMC 17
trames 27
TRANSIT 16
trap_sgn 40
Triton X100 14
TS 73

tterm 47
tuyau 87
UART 19; 27; 29
UNIX 25
vannes 53; 87
variable 45
vidange 88
WAECO 94

