

ORSTOM

MONTPELLIER

Année 1993

Cellule Hydrologie

RAPPORT DE STAGE

EMULATEUR TD91

Matériel : PC

Langage : Turbo C

Réalisé par :

BAEZA Yvan

LUGUEL Olivier

REMERCIEMENTS

REMERCIEMENTS

L'élaboration de ce projet a été effectué au sein du Laboratoire d'Hydrologie du centre ORSTOM de Montpellier.

Il s'inscrit dans le cadre du stage de fin d'année spéciale Informatique et compte pour l'obtention du diplôme universitaire de technologie d'Analyste-Programmeur.

Nous tenons à adresser nos plus vifs remerciements pour leur accueil et leur encadrement à :

- Monsieur M. GAUTIER, ingénieur hydrologue
- Monsieur P.MARCHAN, ingénieur hydrologue
- Monsieur J.P CHAZARIN, technicien hydrologue

Nous remercions également Messieurs MOREL, RAOUS, CHARTIER, DEBUICHE et RABBIA pour leur amabilité et leur disponibilité.

Nos remerciements s'adresse aussi à tous ceux qui ne sont pas cités mais qui ont contribué à la réalisation de ce projet.

INTRODUCTION

INTRODUCTION

Dans le rapport suivant, nous allons détailler toutes les parties qui nous ont amené à l'élaboration du logiciel.

Le travail s'est décomposé en deux temps : Dans le premier nous avons réalisé l'analyse en essayant d'envisager toutes les possibilités de configuration des centrales et de leurs modes de communication et ensuite nous sommes passés au deuxième niveau qui fut celui de la programmation en Turbo C.

Ce rapport va se découper en cinq parties dans lesquelles nous verrons: pour la première, la présentation de l'ORSTOM et de ces activités dans le monde, ensuite dans la deuxième nous définirons le sujet qui nous a été proposé ainsi que son cahier des charges, ce qui nous emmènera à développer la troisième partie dans laquelle il sera question de l'analyse qui fut réalisée en développant les trois types de centrales, les modes de communication ainsi que la gestion logicielle de la RS232, ces trois parties nous préparerons à la quatrième qui développe toutes les facettes de la programmation.

Ensuite dans la cinquième partie nous ferons un guide à l'utilisateur et nous expliquerons les différentes fenêtres que l'utilisateur aura à visualiser ainsi que les différentes fonctions et leurs utilisations.

SOMMAIRE

SOMMAIRE

REMERCIEMENTS

INTRODUCTION

| | |
|--|----------------|
| CHAPITRE I PRESENTATION DE L'ORSTOM | Page 1 |
| Centre ORSTOM de MONTPELLIER | Page 2 |
| Laboratoire d'Hydrologie | Page 6 |
| | |
| CHAPITRE II DEFINITION DU SUJET | Page 11 |
| Projet | Page 11 |
| Cahier des charges | Page 12 |
| | |
| CHAPITRE III ANALYSE | Page 14 |
| III/1 INTRODUCTION | Page 14 |
| III/2 LES TYPES DE CENTRALE | Page 15 |
| III/3 MODE DE COMMUNICATION | Page 17 |
| | |
| III/4 GESTION DE LA LIAISON RS232 | Page 18 |
| III/4.1 Introduction | Page 18 |
| III/4.2 Les commandes | Page 18 |
| III/4.3 Remarques | Page 30 |
| | |
| III/5 LES LIBRAIRIES | Page 31 |
| III/5.1 Introduction | Page 31 |
| III/5.2 Détails sur les librairies | Page 31 |
| - Librairie SERIAL .H | Page 32 |
| * La gestion des fenêtres | Page 33 |
| * La gestion du clavier | Page 38 |
| * Les fonctions | Page 43 |
| - Librairie CONVERT .H | Page 46 |
| - Librairie GRAPHIQUE .H | Page 48 |

| | | |
|--|----------------------------------|---------|
| III/6 | GRAPHSETS | Page 50 |
| III/6.1 | Introduction | Page 50 |
| III/6.2 | Détails | Page 51 |
| CHAPITRE IV PROGRAMMATION | | Page 54 |
| IV/1 | INTRODUCTION | Page 54 |
| * Librairie | SERIAL .H | Page 55 |
| * Librairie | GRAPHIQUE .H | Page 78 |
| * Librairie | CONVERT .H | Page 81 |
| * Librairie | DISK .H | Page 84 |
| * Librairie | TEXTE .H | Page 86 |
| * Programme principal | TRM1 .CPP | Page 89 |
| CHAPITRE V GUIDE DE L'UTILISATEUR | | Page 93 |
| I | / INTRODUCTION | Page 93 |
| II | / MATERIEL | Page 93 |
| III | / INSTALLATION | Page 93 |
| IV | / LANCEMENT DU LOGICIEL | Page 93 |
| V | / DEROULEMENT DU LOGICIEL | Page 94 |
| V/1 | Ecran de présentation | Page 94 |
| V/2 | Transfert | Page 95 |
| V/3 | Visualisation des paramètres | Page 96 |
| V/4 | Les fenêtres | Page 98 |
| | V/4.1 Introduction | Page 98 |
| | V/4.2 Visualisation des fenêtres | Page 99 |
| V/5 | Les commandes | Page106 |
| | V/5.1 Introduction | Page106 |
| | V/5.2 Détails sur les commandes | Page107 |
| CONCLUSION | | Page121 |
| ANNEXES | | |

CHAPITRE I

PRESENTATION DE L' O.R.S.T.O.M

CHAPITRE I PRESENTATION DE L'ORSTOM

L'INSTITUT Français de Recherches Scientifiques...

L'ORSTOM est un établissement public à caractère scientifique et technologique placé sous la tutelle des Ministres chargés de la Recherche et de la Coopération.

Depuis près de cinquante ans, il conduit des recherches sur les milieux intertropicaux qui sont devenus des références internationales.

...pour le développement...

Il propose à ses partenaires du Sud et aux acteurs du développement des programmes et des résultats sur quatre grands thèmes:

- * environnement et grands écosystèmes océaniques, aquatiques et terrestre
- * agriculture en milieux tropicaux fragiles
- * environnement et santé (SIDA, ...)
- * homme et sociétés en mutations

... en Coopération

Ces recherches sont conduites en coopération avec des institutions du Sud, en fonction des choix scientifiques et techniques. L'ORSTOM participe au renforcement des capacités scientifiques du Sud par la formation à la recherche et par des appuis spécifiques.

CENTRE ORSTOM DE MONTPELLIER

Il comprend:

*** Cinq départements pluridisciplinaires**

Ils élaborent et mettent en oeuvre les programmes qui sont exécutés par des équipes relevant de quarante et une unités regroupées dans ces départements :

- Terre, Océan, Atmosphère
- Eaux Continentales
- Milieux et Activités Agricoles
- Société, Urbanisation, Développement

*** Sept commissions scientifiques**

Elles évaluent régulièrement le déroulement des programmes et la carrière des chercheurs qui les exécutent.

Elles regroupent plus de quarante disciplines sous leurs intitulés:

- géologie, géophysique
- hydrologie, pédologie
- hydrobiologie et océanographie
- science du monde végétal
- science biologique et biochimique appliquée à l'homme
- science sociale
- science de l'ingénieur et de la communication

Photographie du Centre



Coopération Scientifique

Sur les 1500 chercheurs, ingénieurs et techniciens relevant des départements scientifiques de L'ORSTOM en 1993 :

** 600 sont en affectation de longue durée en Afrique, en Amérique Latine et dans le Pacifique dont 400 dans des Institutions du Sud, Nationales, Régionales ou Intercontinentales.

** 500 sont des techniciens originaires des pays du Sud.

** Une cinquantaine de chercheurs étrangers sont directement associés par contrat au programme de recherche de l'Institut.

Grâce à un partenariat très actif avec des institutions scientifiques des pays du Nord et particulièrement sa participation à de grands programmes internationaux, l'ORSTOM permet la concentration de connaissances et la compétence sur des problèmes spécifiques de la recherche dans les régions chaudes.

Le soutien aux communautés et aux réseaux scientifiques des pays du Sud

Ce soutien est assuré par la réalisation en commun de programmes de recherche mais aussi :

- par le transfert aux autorités nationales et la transformation des centres ORSTOM d'Afrique Sub-Saharienne.

-par la création et la gestion en commun d'installations et de laboratoires.

- par l'appui financier, technologique, documentaire et informatique aux chercheurs et aux institutions partenaires pour l'exécution de leurs programmes.

Laboratoire Hydrologique

Le laboratoire en collaboration étroite avec les centres ORSTOM implantés en territoires Outre-Mer étudie et développe des programmes et des systèmes pour l'acquisition, le traitement et l'interprétation de mesures pluviométriques basées sur l'émission et la réception satellite à l'aide de matériel au sol.

Le matériel est testé dans la région Languedocienne avant son envoi sur son site final à l'étranger.

Les dispositifs au sol sont les suivants :

- **Centrales d'acquisitions** : Pluviométriques ou Limnimétriques
(ARGOS ou METEOSAT)

- **Pluviomètres** : Basculeurs à augets
Capteurs de Pression (SPI)

- **Divers** : Sondes de température
Sondes de luminansité (ensoleillement)
Anémomètres
etc...

Descriptions succinctes des différents dispositifs :

- Centrales d'Acquisitions :

Nous avons surtout utilisé les centrales à télétransmission par satellites à défilement ARGOS.

Ces centrales assurent l'acquisition des données hydrométéorologiques (hauteur et température d'eau, pluviométrie)

Associées à une station ARGOS, elles permettent de constituer des réseaux complets de télésurveillance de bassins fluviaux, de retenues d'eau, etc...

Les plates-formes METEOSAT assurent les mêmes types d'acquisition que les plates-formes ARGOS, mais utilisent pour les transmissions les satellites géostationnaires METEOSAT, GOES, GMS...

Pluviomètres :

- Basculeurs à Auget :

L' ORSTOM utilise beaucoup de matériel développé par la société ELSYDE qui propose un basculeur à auget à coupleur à fenêtres (optocoupleurs)

Ce système fonctionne sur le principe de basculement, en effet lorsqu'il pleut, l'eau est recueillie par un entonnoir dont on peut modifier la taille pour une précision plus ou moins importante, cette eau remplit une coupelle (Auget) et lorsque celle-ci est presque pleine, par effet de balance, il y a basculement et c'est la deuxième coupelle qui se remplit à son tour. Le basculement a eu pour effet de vider la coupelle et d'envoyer via le photocoupleur une impulsion vers la centrale ou elle sera enregistrée et datée.

Photographie :

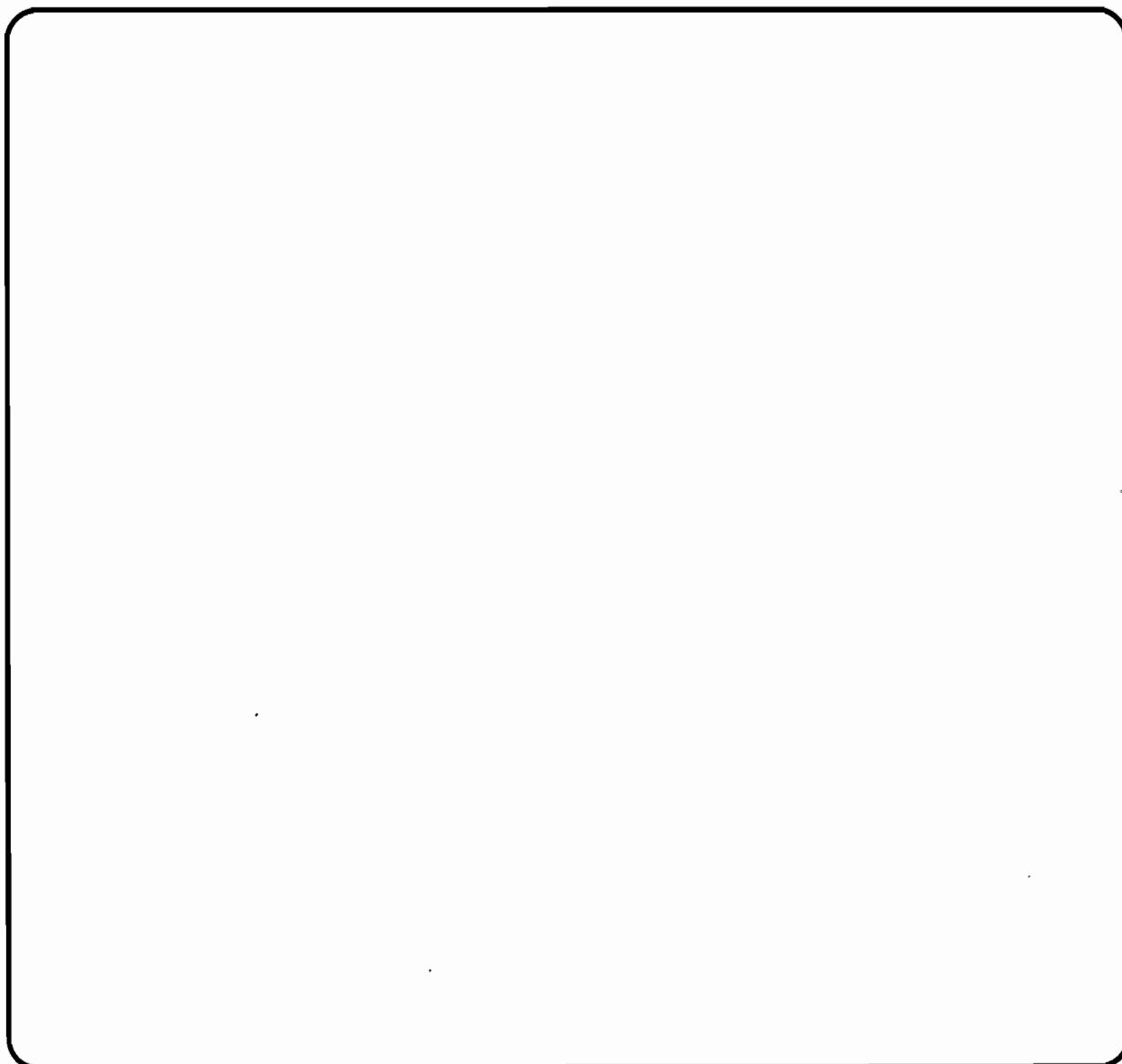


Capteurs de Pression (SPI) :

Développé en collaboration avec les services hydrologiques de l'ORSTOM, le SPI est un ensemble électronique et informatique destiné à la mesure précise de niveaux d'eau et de la température des liquides.

Ce capteur est très précis, mais nécessite pour un fonctionnement optimal une immersion d'au moins 40 mm.

Photographie :



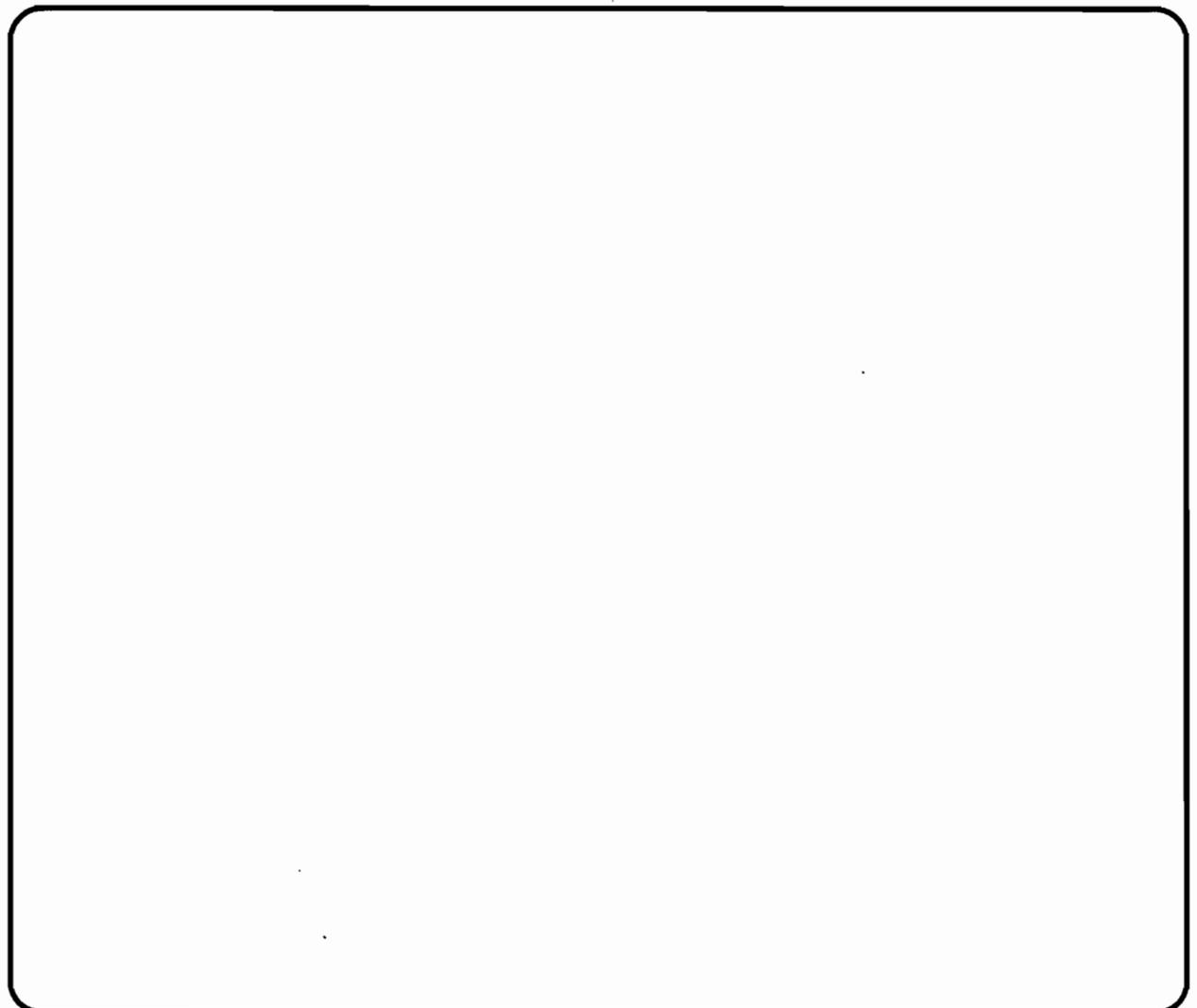
Terminaux de Terrain :

Lorsque les plates-formes sont installées sur site, il est nécessaire de les programmer et de vérifier leur bon fonctionnement.

Ce travail est réalisé à l'aide de terminaux de terrain prééglés à la programmation.

L'utilisateur n'a qu'à brancher ce dernier sur la centrale et le TD se charge de toutes les opérations, l'utilisateur se trouve devant un écran LCD, et n'a qu'une dizaine de touches à utiliser. Avec ce TD, il va régler la date et l'heure de la centrale, vérifier le fonctionnement de basculeur à auget, faire des mesures avec le SPI, régler le seuil et la période de déclenchement des mesures du SPI, reformater les EEPROMS, ETC..

Photographie :



CHAPITRE II

DEFINITION DU SUJET

CHAPITRE II DEFINITION DU SUJET

- **Projet :**

L'ORSTOM utilise pour l'acquisition des données pluviométriques des centrales pouvant être définies sous trois types différents (Créf Chapitre III -TYPE DE CENTRALE-).

Pour la programmation et la vérification de ces dernières, on utilise un appareil appelé terminal de terrain ou TD (Créf Chapitre I -LABORATOIRE-).

L'ORSTOM trouve cet appareil comme étant révolu et désire utiliser un ordinateur portable pour toutes ces opérations. Il nous a donc été demandé de créer un programme qui à l'aide d'un PC et via la liaison RS232 pouvait remplacer le TD.

La liaison RS232 a été utilisée car le TD et la centrale communiquait déjà sous ce protocole, donc aucune modification n'a été apporté de ce côté là.

- Cahier des charges :

- Modulable et Réutilisable

Le programme devant s'intégrer par la suite dans un logiciel existant, il a fallu utiliser le langage de ce logiciel, donc :

Langage : TURBO C

Matériel : PC 286

- Facilité d'emploi :

Il faut garder les mêmes types d'écrans qu'avec le TD de façon à ne pas dépayser les utilisateurs.

-Compatibilité avec un PC portable :

Le choix des couleurs et du type d'écran fut important, car le logiciel sera utilisé sur des portables à écrans monochrome ou plasma dont les jeux de couleurs sont limités.

- Adaptable :

Le programme doit tenir compte du type de centrale que l'on est en train d'interroger pour n'afficher que les fenêtres nécessaires à cette centrale ce qui est un plus appréciable pour les utilisateurs qui n'ont à leur porté que les éléments nécessaires.

- Nouvelle Configuration :

Au lieu d'avoir une fenêtre après l'autre, faire apparaître le maximum d'informations sur le minimum d'écran.

- Structuré et Retranscrivable :

Tous les messages écran doivent constituer une librairie bien distincte de façons à pouvoir récrire cette librairie dans d'autres langues pour que le logiciel puisse êtres utilisé dans le Monde tout en ayant un minimum de connaissances informatiques.

CHAPITRE III

ANALYSE

CHAPITRE III ANALYSE

III/1 INTRODUCTION

Avant de commencer la programmation, il nous a fallu développer une analyse assez complète qui a eu pour but de dissocier les 3 types de centrales, les différents modes de communication et la gestion de la liaison RS232 .

Cette dernière partie fût celle qui nous a pris le plus de temps car les centrales nécessitent pour les interroger et les programmer une syntaxe de communication bien définie .

Les problèmes sont venus de la création même des centrales, en effet celles-ci ont été développées par la société ELSYDE, mais celle-ci ne pense pas au travail qui doit être mis en oeuvre par la suite, car il faut interroger la centrale soit en 8 bits de datas, soit en 7 bits selon la commande, il faut envoyer ou réceptionner des caractères soit en Héxa, soit en BCD, soit en ASCII.

Mais toutes ces différences n'étaient pas stipulées dans aucune documentation et nous avons dû les découvrir au fur et à mesure de notre progression dans la programmation. De plus dans les documentations sur la syntaxe, à utiliser pour communiquer avec les centrales, l'ordre et la signification des octets reçus n'a rien a voir avec la réalité, d'où des problèmes que nous avons rencontré lors de la mise en oeuvre du logiciel

Tous ces problèmes nous ont retardé dans l'analyse et ont porté un préjudice sur la programmation .

III/2 LES TYPES DE CENTRALE

Nous avons donc vu qu'il existe 3 types de centrales qui ont pour rôle l'acquisition des données pluviométriques.

**** Centrale PLUVIO :**

Elle est conçue pour être connectée à un pluviomètre à auget basculeur équipé d'un contact mercure, ou "red", ou opto-électronique .
Le support de mémorisation est constitué par 2 mémoires fixes de 32 KOctets chacune, effaçables électriquement (EEPROMS) ...

Ce support statique non volatile est particulièrement bien adapté aux environnements sévères .

Cette capacité de 64 KOctets permet l'enregistrement de 20 000 basculements .

**** Centrale LIMNI :**

Elle est conçue pour être connectée à une sonde limnimétrique du type SPI .

Les autres caractéristiques sont les mêmes que pour une centrale PLUVIO, mais les 64 KOctets de mémoire permettent de stocker 9 400 mesures provenant de la sonde .

**** Centrale PLUVIO-LIMNI :**

Celle-ci regroupe toutes les possibilités d'une centrale PLUVIO et d'une centrale LIMNI, mais la capacité mémoire étant la même, celle-ci se trouve partagée comme suit :

- 32 KOctets pour la partition PLUVIO
- 32 KOctets pour la partition LIMNI

Cette différence de mémoire implique donc que l'on ne pourra stocker que :

- 11 000 basculements
- 4 700 mesures .

III/3 MODE DE COMMUNICATION :

Les centrales peuvent être interrogées sous 2 configurations d'initialisation du port série .

***** Configuration 1**

- 7 bits de datas
- 1 bit de start
- 1 bit de stop
- 1 bit de parité (paire)

Cette configuration servira uniquement pour une seule commande, (commande "F"), qui sert à connaître l'état des EEPROMS, car la réponse est en ASCII codé sur 7 bits .

***** Configuration 2**

- 8 bits de datas
- 1 bit de start
- 1 bit de stop

Cette configuration sera utilisée pour toutes les autres commandes .

Divers :

- Vitesse : 4 800 bauds
- Liaison : Half Duplex

- Signaux gérés : TX, RX, RTS, CTS ...

III/4 GESTION DE LA LIAISON RS232 :

III/4.1 Introduction

Pour communiquer avec les centrales, nous devons utiliser une syntaxe propre à celles-ci . Nous allons détailler dans la suite de cette partie toute l'analyse réalisée sur l'ensemble des commandes disponibles .

Nous donnerons pour chaque commande à quel type de centrale elle s'adresse, quel est son rôle et la signification de chaque octet .

III/4.2 Les commandes

**** Commande "D"**

- Centrale : toutes
- Utilité : Donne tous les paramètres de fonctionnement
- Détails :

Format de la commande "D" (ASCII)

A cette commande , la centrale renvoie 29 Octets formatés comme suit :

Octet 1 : Identificateur de la centrale (codé HEXA)

- 00 = PLUVIO non ARGOS
- 01 = PLUVIO avec transmission ARGOS
- 02 = PLUVIO avec transmission MODEM
- 04 = LIMNI non ARGOS
- 05 = LIMNI avec transmission ARGOS
- 06 = LIMNI avec transmission MODEM
- 07 = PLUVIO-LIMNI non ARGOS
- 08 = PLUVIO-LIMNI avec transmission ARGOS
- 09 = PLUVIO-LIMNI avec transmission MODEM

Octets 2-3 : N° de série de la centrale (codé HEXA)

L'octet 2 représente le poids fort .

- Octets 4-5 : Cumul total des basculements (codé HEXA)
L'octet 4 représente le poids fort .
- Octets 6-7 : Nombre d'octets utilisés en EEPROM (codé HEXA)
L'octet 6 représente le poids fort .
- Octets 8-13 : Date courante (codée BCD) sur 6 octets dans l'ordre :
Année - Mois - Jour - Heure - Minute - Seconde .
- Octets 14-17 : Date du dernier basculement (codée BCD) sur 4 octets
dans l'ordre:
Mois - Jour - Heure - Minute .
- Octets 18-20 : Date de mise en route (codée BCD) sur 3 octets dans
l'ordre :
Année - Mois - Jour .
- Octets 21-22 : Nombre de jours de fonctionnement (codés HEXA)
L'octet 21 représente le poids fort .
- Octets 23-24 : Nombre d'octets défectueux décelés (codés HEXA)
L'octet 23 représente le poids fort .
- Octet 25 : Tension de la batterie externe servant à l'alimentation de la
balise ARGOS (codée HEXA et en dixièmes de volt) .
- Octet 26 : Tension des piles internes ou de la batterie externe
(codée HEXA et en dixièmes de volt) .
- Octet 27 : Taille de l'auge (codée HEXA) .
- Octet 28 : Pas de temps des cumuls destinés au Message ARGOS
(codé HEXA et en minutes) .
- Octet 29 : Etat du système, chaque bit a une signification propre :
- bit 0 : = 1 si EEPROMS HS
 - bit 1 : = 1 si EEPROMS Pleines
 - bit 2 : = 1 si Enregistrement Bloqué
 - bit 3 : = 1 si Tension pile faible .

**** Commande "F"**

- Centrale : toutes
- Utilité : Etat des EEPROMS
- Détails :

Format de la commande "F" (ASCII)

Cette commande nécessite la configuration 1 pour le port de communication .
De plus le nombre d'octets reçus dépend du type de centrale :

~~ Centrales PLUVIO ou centrale LIMNI :

La centrale renvoie 10 octets formatés comme suit :

- Octets 1-4 : Adresse du dernier octet écrit (en HEXA codé ASCII)
L'octet 1 représente le poids fort .
- Octets 5-8 : Nombre d'octets défectueux (en HEXA codé ASCII)
L'octet 5 représente le poids fort .
- Octets 9-10 : Caractères de fin de transmission .

Exemple : On envoie "F"
 On réceptionne "30" "30" "35" "39" "30" "30" "30" "30" "17" "15"

"30" (HEXA) ==> "0" (ASCII)

"35" (HEXA) ==> "5" (ASCII)

.....

d'où Adresse du dernier octet écrit : 0039 h ==> 57 décimal
 Nombre d'octets défectueux : 0000 h ==> 0 décimal .

~ Centrales PLUVIO-LIMNI :

La centrale renvoie 14 octets formatés comme suit :

- Octets 1-4 : Adresse du dernier octet écrit, partition PLUVIO
(en HEXA codé ASCII)
L'octet 1 représente le poids fort .
- Octets 5-8 : Adresse du dernier octet écrit, partition LIMNI
(en HEXA codé ASCII)
L'octet 5 représente le poids fort .
- Octets 9-12 : Nombre d'octets défectueux (en HEXA codé ASCII)
L'octet 9 représente le poids fort .
- Octets 13-14 : Caractères de fin de transmission .

**** Commande "M"**

- Centrale : toutes
- Utilité : Permet de faire une mesure SPI
- Détails :

Format de la commande "M" (ASCII)

La réponse de la centrale comporte 6 octets formatés comme suit :

- Octets 1-2 : Température (codée HEXA et en dixièmes de degré)
L'octet 1 représente le poids fort .
- Octets 3-4 : Hauteur (codée HEXA et en millimètres)
L'octet 3 représente le poids fort .
- Octets 5-6 : N° du SPI (codé BCD)
L'octet 6 représente le début du N°
L'octet 5 représente la fin du N° .

**** Commande "G"**

- Centrale : LIMNI et PLUVIO-LIMNI
- Utilité : Transfert du bloc de formatage des EEPROMS
- Détails :

Format de la commande : "G" (ASCII)

La centrale renvoie 6 octets formatés comme suit :

- Octet 1 : Seuil d'enregistrement (codé HEXA et en centimètres)
- Octet 2 : Période de mesure (codée HEXA et en minutes)
- Octet 3-4 : Décalage Hauteur (codé HEXA et en millimètres)
L'octet 3 représente le poids fort .
- Octets 5-6 : Adresse du dernier octet écrit (codé HEXA)
L'octet 5 représente le poids fort .

**** Commande "I"**

- Centrale : LIMNI et PLUVIO-LIMNI
- Utilités : Permet le formatage des EEPROMS
- Détails :

Format de la commande "I" (ASCII)

La centrale renvoie "I", pour OK prêt à recevoir, alors on envoie 4 octets formatés comme suit :

- Octet 1 : Seuil d'enregistrement (codé HEXA et en centimètres)
- Octet 2 : Période de mesure (codée HEXA et en minutes)
- Octets 3-4 : Décalage hauteur (codé HEXA et en millimètres)
L'octet 4 représente le poids fort .

La centrale renvoie "O", pour OK bien enregistré.

**** Commande "Y"**

- Centrale : PLUVIO et PLUVIO-LIMNI
- Utilités : Modification de la taille de l'auget
- Détails :

Format de la commande "Y" (ASCII)

La centrale renvoie "Y", pour OK prêt à recevoir, puis l'on envoie 1 octet :

Octet 1 : Taille de l'auget (codée HEXA et en dixièmes de millimètres).

La centrale renvoie "O", pour OK bien enregistré .

**** Commande "R"**

- Centrale : toutes
- Utilité : Réinitialisation de l'enregistrement
- Détails :

Format de la commande "R" (ASCII)

Cette commande exécute les mêmes opérations que lors de la mise sous tension de la centrale .

Après l'envoi de cette commande, la centrale met environ 3 secondes pour se réinitialiser .

**** Commande "E"**

- Centrale : toutes
- Utilité : Effacement des données enregistrées en EEPROMS
- Détails :

Format de la commande "E" (ASCII)

Après l'envoi de cette commande, la centrale renvoie soit :

- "O" pour Effacement Correct
- "N" pour Effacement Incorrect

La durée de l'effacement peut aller jusqu'à 3 minutes si les EEPROMS sont pleines .

Après cette commande il faut réinitialiser la centrale pour que l'enregistrement recommence

**** Commande "A"**

- Centrale : Toutes, uniquement si elles sont équipées de la transmission ARGOS
- Utilité : Programmation du pas de temps des cumuls
- Détails :

Format de la commande "A" (ASCII)

Après l'envoi de la commande, il faut envoyer 1 octet :

Octet 1 : Pas de temps (codé HEXA et en minutes)

Après réception du pas de temps, la centrale renvoie "O", pour OK bien enregistré .

**** Commande "H"**

- Centrale : toutes
- Utilité : Réglage de l'horloge
- Détails :

Format de la commande "H" (ASCII)

La centrale renvoie "H", pour OK prêt à recevoir, puis l'on envoie 6 octets formatés comme suit : (tous les octets sont codés BCD)

| | |
|---------|-----------|
| Octet 1 | : Année |
| Octet 2 | : Mois |
| Octet 3 | : Jour |
| Octet 4 | : Heure |
| Octet 5 | : Minute |
| Octet 6 | : Seconde |

après réception de ces 6 octets, la centrale renvoie soit :

"O" pour Horloge correctement programmée

"N" pour Horloge incorrectement programmée
ou pour Horloge en panne .

**** Commande "B"**

- Centrale : toutes
- Utilité : Début de test de l'auget
- détails :

Format de la commande "B" (ASCII)

Lorsque la centrale reçoit cette commande, elle bloque l'enregistrement et elle renvoie, pour chaque basculement de l'auget :

"O" pour impulsion reçue .

**** Commande "Z"**

- Centrale : toutes
- Utilité : Fin de test de l'auget
- Détails :

Format de la commande "Z" (ASCII)

Lorsque la centrale reçoit cette commande, elle repasse en mode enregistrement .

**** Commande "P"**

- Centrale : PLUVIO et PLUVIO-LIMNI
- Utilité : Calcul de la pluie tombée à une date précise
- Détails :

Format de la commande "P" (ASCII)

On envoie ensuite 3 octets formatés comme suit :

| | | | | | |
|---------|---|-------|---|-----------|---|
| Octet 1 | : | Année | (| codée BCD |) |
| Octet 2 | : | Mois | (| codé BCD |) |
| Octet 3 | : | Jour | (| codé BCD |) |

Après réception de ces 3 octets, la centrale recherche, dans les EEPROMS, la date demandée .

Si aucune pluie n'est tombée ce jour la, elle recherche la date la plus proche.

La réponse de la centrale comporte 5 octets formatés comme suit :

| | | | | | |
|------------|---|-------|---|-----------|---|
| Octet 1 | : | Année | (| codée BCD |) |
| Octet 2 | : | Mois | (| codé BCD |) |
| Octet 3 | : | Jour | (| codé BCD |) |
| Octets 4-5 | : | Cumul | (| codé HEXA |) |

L'octet 4 représente le poids fort .

Si aucune donnée n'est enregistrée, la centrale renvoie, pour chaque octet, "0" .

**** Commande "N"**

- Centrale : PLUVIO et PLUVIO-LIMNI
- Utilité : Pluie du jour suivant
- Détails :

Format de la commande "N" (ASCII)

Cette commande ne sera prise en compte par la centrale que si elle a été précédée par une commande "P" ou par une commande "J" ou par elle-même .

Elle permet de trouver le cumul de pluie pour le jour immédiatement suivant du jour en cours .

La réponse de la centrale est identique à celle de la commande "P"

**** Commande "J"**

- Centrale : PLUVIO et PLUVIO-LIMNI
- Utilité : Pluie du jour précédent
- Détails :

Format de la commande "J" (ASCII)

De la même manière, cette commande ne sera prise en compte que si elle a été précédée par une commande "P" ou par une commande "N" ou par elle-même .

Elle permet d'obtenir le cumul de pluie pour le jour immédiatement précédent le jour en cours .

La réponse de la centrale est identique à celle de la commande "P"

**** Commande "L"**

- Centrale : PLUVIO et PLUVIO-LIMNI
- Utilité : Pluie du jour transmise heure par heure
- Détails :

Format de la commande "L" (ASCII)

Cette commande s'intercale après les commandes "P", "N" ou "J" .
Elle permet d'obtenir, pour le jour en cours, les cumuls détaillés heure par heure .

La réponse de la centrale est d'une longueur variable multiple de 4 .

Pour chaque heure où la pluie est tombée, la centrale transmet 4 octets formatés comme suit :

| | | |
|------------|--------------------------|----------------|
| Octet 1 | : Heure de début | (codée HEXA) |
| Octet 2 | : Heure de fin | (codée HEXA) |
| Octets 3-4 | : Nombre de basculements | (codés HEXA) |

L'octet 3 représente le poids fort .

La fin de transmission est repérée par 4 octets tous égaux à "O"

III/4.3 Remarques

Nous venons donc de voir les différentes commandes, ainsi que leurs syntaxes, qui sont nécessaires pour programmer ou pour lire les paramètres des différentes centrales .

Le programme qui a été réalisé utilise toutes ces commandes mais de façon implicite pour l'utilisateur, en effet celui-ci n'aura jamais à taper les octets l'un après l'autre ce qui ne serait pas convivial mais qui de plus ne fonctionnerait pas à cause du temps qui est impartie pour chaque transmission .

Les difficultés rencontrées ont porté essentiellement sur les concordances de type pour les variables (HEXA, INTEGER, FLOAT, ASCII, BCD,) .

Le Turbo C ne fournissant pas tous les utilitaires permettant de passer d'un type à un autre, nous avons dû réaliser une librairie CONVERT .H dans laquelle sont regroupées toutes les fonctions de conversion dont nous avons eu besoin et qu'il nous a fallu créer .

De plus pour gérer la RS232, il nous a fallu développer des fonctions permettant l'initialisation du port de communication selon la configuration choisie, la réception d'un caractère, l'envoi d'un caractère sur la ligne, la réception dans un tableau, la vérification de l'état de la ligne, etc ..., ces fonction sont regroupées dans la librairie SERIAL .H .

Toutes les librairie seront expliquées dans la suite de ce chapitre .

III/5 LES LIBRAIRIES :

III/5.1 Introduction

Le programme se découpe en 3 parties bien distinctes . Dans la première nous avons toutes les fonctions de gestion de la RS232 (Librairie SERIAL .H), toutes les fonctions de gestion du mode graphique (Librairie GRAPHIQUE .H), toutes les fonctions de conversion de type (Librairie CONVERT .H) .

La seconde partie porte sur la gestion des fenêtres d'affichage (Librairie SERIAL .H) tandis que la troisième s'oriente sur la gestion du clavier (Librairie SERIAL .H) .

Le programme tourne donc autour de :

- La réception de 4 tableaux provenant des centrales lors de leurs interrogations
- La gestion des 12 fenêtres d'affichage
- La gestion des 12 commandes clavier .

L' affichage d'une telle ou telle fenêtre dépend essentiellement du type de la centrale, mais dépend aussi de l'état du système qui sera connu en interprétant l'octet 29 de la commande "D", en effet si celui-ci est différent de 0 alors il sera impossible de modifier les paramètres de la centrale .

III/5.2 Détails sur les librairies

Cette partie sera découpée comme suit :

- ** Librairie SERIAL .H**
 - La gestion des fenêtres
 - La gestion du clavier
 - Les fonctions

- ** Librairie CONVERT .H**
 - Les fonctions

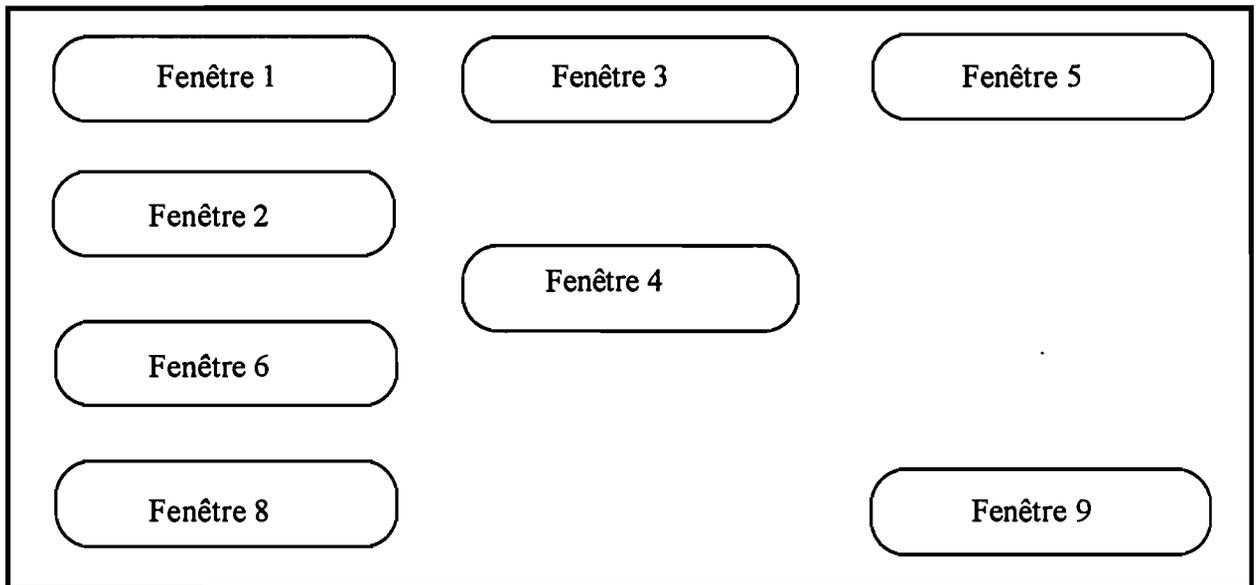
- ** Librairie GRAPHIQUE .H**
 - Les fonctions

LIBRAIRIE SERIAL .H

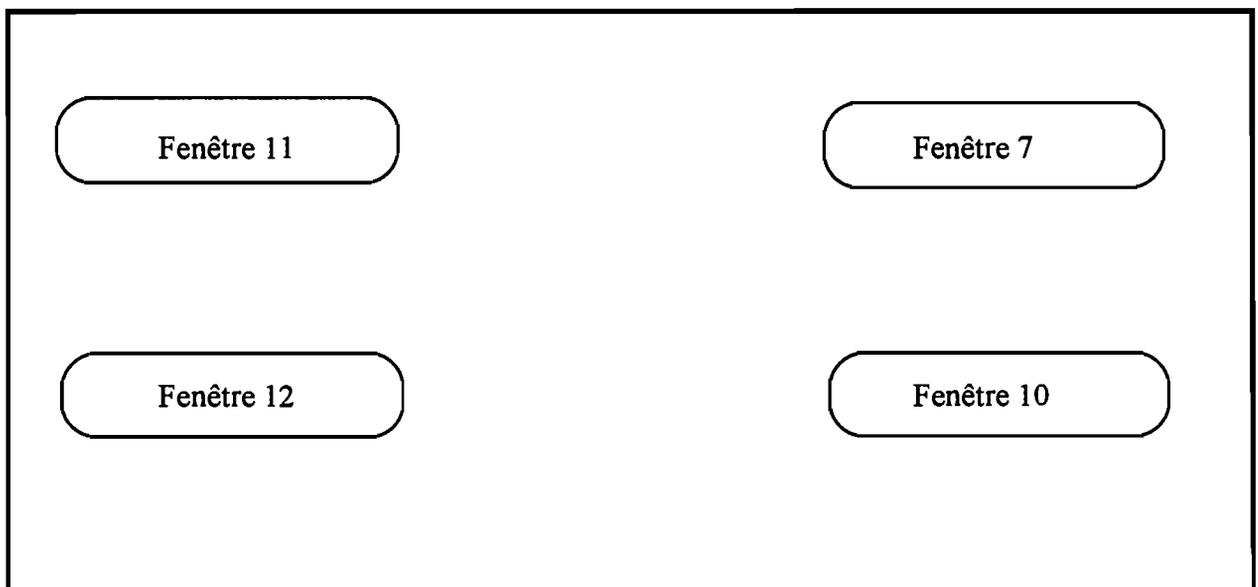
***** La gestion des fenêtres :**

Le nombre des fenêtres étant trop important pour que celles-ci puissent apparaître sur un seul écran, il nous a fallut décomposer l'affichage sur 2 écrans .

Ecran1 :



Ecran2 :



Description des différentes fenêtres :

**** Fenêtre 1**

- Centrale : toutes
- Utilité : Affiche le type de la centrale
- Détails :

Cette fenêtre utilise l'octet 1 de "D" pour le type de la centrale, puis les octets 2 et 3 de "D" pour le N° de série .

**** Fenêtre 2**

- Centrale : toutes
- Utilité : affichage de la date et de l'heure
- Détails :

Cette fenêtre utilise les octets 8 à 13 de "D" pour déterminer la date et l'heure, ensuite on appelle la fonction "Pendule()" qui sert à synchroniser l'horloge PC avec l'horloge Centrale . Ainsi nous obtenons un affichage de l'heure en temps réel .

**** Fenêtre 3**

- Centrale : toutes
- Utilité : Etat des EEPROMS
- Détails :

Cette fenêtre utilise les octets 25 et 26 de "D" pour définir la valeur de la tension des batteries et des piles .

**** Fenêtre 4**

- Centrale : toutes
- Utilité : Traitement de la durée de fonctionnement
- Détails :

Cette fenêtre utilise les octets 18 à 22 de "D" pour donner le jour de début de fonctionnement de la centrale ainsi que le nombre de jours écoulés depuis .

**** Fenêtre 5**

- Centrale : toutes
- Utilité : Affichage de la taille de l'auget
- Détails :

Cette fenêtre utilise l'octet 27 de "D" pour donner la valeur de la taille de l'auget .

**** Fenêtre 6**

- Centrale : PLUVIO et PLUVIO-LIMNI
- Utilité : Affichage du dernier basculement et du cumul
- Détails :

Cette fenêtre utilise les octets 14 à 17 de "D" pour définir la date du dernier basculement, ainsi que les octets 4 et 5 de "D" pour la valeur du cumul .

**** Fenêtre 7**

- Centrale : PLUVIO et PLUVIO-LIMNI
- Utilité : Affichage de l'état des EEPROMS (partition PLUVIO)
- Détails :

Cette fenêtre utilise les octets 1 à 4 de "F" pour définir le nombre d'octets enregistrés, ainsi que les octets 5 à 8 de "F" pour une centrale PLUVIO ou les octets 9 à 12 de "F" pour une centrale PLUVIO-LIMNI afin de définir le nombre d'octets défectueux .

**** Fenêtre 8**

- Centrale : toutes
- Utilité : Affichage du test de l'auge
- Détails :

Cette fenêtre sert uniquement à l'affichage du compteur de basculements lorsque l'on est en mode de test .

**** Fenêtre 9**

- Centrale : LIMNI et PLUVIO-LIMNI
- Utilité : Affichage des paramètres de formatage
- Détails :

Cette fenêtre utilise l'octet 1 de "G" pour afficher le seuil de déclenchement d'une mesure SPI, les octets 3 et 4 de "G" pour afficher le décalage à apporter aux mesures de hauteur d'eau ainsi que l'octet 2 de "G" afin de donner la période entre 2 mesures .

On affiche aussi les octets 5 et 6 de "M" de façon à connaître le N° du SPI qui est connecté à la centrale .

**** Fenêtre 10**

- Centrale : LIMNI et PLUVIO-LIMNI
- Utilité : Affichage de l'état des EEPROMS (partition LIMNI)
- Détails :

Cette fenêtre utilise les octets 1 à 4 de "F" pour une centrale LIMNI ou les octets 5 à 8 de "F" pour une centrale PLUVIO-LIMNI afin de définir le nombre d'octets enregistrés, ainsi que les octets 5 à 8 de "F" pour une centrale LIMNI ou les octets 9 à 12 de "F" pour une centrale PLUVIO-LIMNI afin de définir le nombre d'octets défectueux .

**** Fenêtre 11**

- Centrale : LIMNI et PLUVIO-LIMNI
- Utilité : Affichage de la température et de la hauteur d'eau
- Détails :

Cette fenêtre utilise les octets 5 et 6 de "M" pour afficher le N° du SPI, ensuite les octets 2 et 1 pour la température et les octets 3 et 4 pour la hauteur d'eau .

**** Fenêtre 12**

- Centrale : toutes, uniquement si transmission ARGOS
- Utilité : Affichage du pas de temps des cumuls
- Détails :

Cette fenêtre utilise l'octet 28 de "D" pour afficher la valeur du pas de temps de la transmission au satellite ARGOS .

*** La gestion du clavier :

Lors du lancement du programme, nous avons à notre disposition un choix de 12 commandes servant chacune à un traitement bien précis .

Ce choix est affiché en bas de l'écran de la façon suivante :

| | | |
|-----------------------|-----------------------------|----------------------------|
| 'Q'uitter | 'E'ffacement EEPROMS | 'H'istogramme |
| 'N'ext | 'A'ujet Taille | 'T'est Auget |
| 'P'récedent | 'M'essage ARGOS | 'F'ormatage EEPROMS |
| 'S' Mesure SPI | 'D'ate et Heure | 'T'initialisation |

Nous retrouvons ici les possibilités qu'avait le TD classique auxquelles nous avons rajouté quelques autres pour que le programme soit plus convivial .

De plus, lorsque l'on sélectionne une commande celle-ci passe en rouge afin de bien visualiser notre choix .

Si l'on sélectionne une commande et qu'elle ne correspond pas à la centrale connectée, rien ne se passe .

**** Commande "Q"**

- Centrale : toutes
- Détails :

A l'appuie sur cette commande, le programme prend fin et l'on retourne au DOS .

**** Commande "N"**

- Centrale : toutes
- Détails :

Cette commande permet de passer à l' Ecran2 .

**** Commande "P"**

- Centrale : toutes
- Détails :

Inversement à la commande "N", celle-ci permet de passer à l' Ecran1 .

**** Commande "S"**

- Centrale : toutes
- Détails :

Cette commande lance le processus de mesure du SPI . On va demander à la centrale de nous transmettre le résultat de la mesure que l'on va stocker dans le tableau [M] .

Ensuite on relancera l'affichage par la fonction " Detect() " ...

**** Commande "E"**

- Centrale : toutes
- Détails :

Cette commande sert à lancer le processus d'effacement des EEPROMS, cette commande ne sera active que si la tension de pile n'est pas trop faible .

La durée du processus peut varier jusqu'à 3 minutes, mais ne peut excéder ce temps .
En fin de processus on relance l'affichage avec la fonction " Detect() " ..

**** Commande "A"**

- Centrale : toutes
- Détails :

Cette commande permet de modifier la taille de l'auget .

**** Commande "M"**

- Centrale : toutes, uniquement si transmission ARGOS
- Détails :

Cette commande permet de modifier la pas de temps d'une mesure destinée au satellite ARGOS .

Le processus ne sera engagé que si la tension des piles n'est pas trop faible .

**** Commande "D"**

- Centrale : toutes
- Détails :

Cette commande permet de modifier la date et l'heure de la centrale .
En fin de processus, on réinitialise l'horloge PC avec l'horloge centrale .

Ce processus ne peut avoir lieu que si la tension des piles est correcte .

**** Commande "H"**

- Centrale : PLUVIO et PLUVIO-LIMNI
- Détails :

Cette commande permet de visualiser, pour un jour précis, le cumul d'eau tombée .
On peut également passer d'un jour à un autre par les macro-commandes "P" et "S", mais l'on peut aussi visualiser le cumul heure par heure par la macro-commande "H" ...

**** Commande "T"**

- Centrale : toutes
- Détails :

Cette commande permet de visualiser les basculements de l'auget pour tester si celui-ci fonctionne correctement .

**** Commande "F"**

- Centrale : LIMNI et PLUVIO-LIMNI
- Détails :

Cette commande sert à rentrer les paramètres de fonctionnement de la sonde SPI, le décalage, le seuil de déclenchement et la période de mesure .

**** Commande "I"**

- Centrale : toutes
- Détails :

Cette commande permet de relancer le processus d'initialisation .
On interroge la centrale et l'on stocke les informations reçues dans les tableaux [D], [M], [F], et [G].

Cette commande permet donc de visualiser si toutes les modifications apportées ont bien été enregistrées par la centrale .

*** Les fonctions :

Nous venons donc de voir les principales fonctions de cette librairie, les listings seront donnés dans le chapitre IV .

Le programme nécessite également plusieurs autres fonctions que nous allons expliquer ci-dessous .

▣ Fonction detect()

Cette fonction sert , lors de l'affichage, à faire apparaître les fenêtres qui correspondent à la centrale connectée .

Cette fonction attend 1 paramètre, pour définir quel est l'écran qu'il faut afficher, tel que :

- 'P' pour Ecran1
- 'N' pour Ecran2 .

▣ Fonction MAKE_NAME()

Cette fonction est utilisée pour créer un fichier portant comme nom le N° de série de la centrale et comme extension le type de celle-ci .

Exemple :

Soit une centrale dont le N° est "18008" et dont le type est "PLUVIO"
Le fichier créé aura pour nom : 18008 .PLU

Ce fichier contiendra tous les tableaux concernant la centrale .

▣ **Fonction Initialiser_Mesure()**

Celle-ci sert à initialiser la liaison série sous la configuration 1 .

▣ **Fonction Initialiser_Port()**

Celle-ci sert à initialiser la liaison série sous la configuration 2 .

▣ **Fonction Envoyer_Caractere()**

Cette fonction sert à envoyer un caractère sur le port série .

▣ **Fonction Regard_Buffer()**

Celle-ci retourne 0 si il y a un caractère disponible sur le port série .

▣ **Fonction Reception_H()**

Cette fonction sert à réceptionner une série d'octets et à les stocker dans le tableau qui aura été passé en paramètre .

La fin de réception s'effectue lorsque l'on reçoit "O".

▣ **Fonction Reception()**

Cette fonction est identique à la précédente sauf que la fin de réception est définie par un nombre qui sera passé comme paramètre d'entrée de cette fonction .

▣ **Fonction Initialisation()**

Cette fonction est la première utilisée, elle sert à demander à la centrale tous ces paramètres de fonctionnement à l'aide des commandes "D", "F", "M" et "G" et à stocker les réponses dans les tableaux respectifs [D], [F], [M] et [G] .

▣ **Fonction Boucle_programme()**

Cette fonction est le centre du programme, c'est elle qui gère toutes les autres fonctions et qui lance le processus complet .

LIBRAIRIE CONVERT .H

▣ **Fonction Pendule()**

Cette fonction sert à récupérer l'horloge PC et à l'afficher sur l'écran .

▣ **Fonction INTFLOAT()**

Cette fonction transforme un tableau d'entiers en un tableau de nombres flottants .
Ceci est utilisé lors de l'affichage de l'état des EEPROMS .

▣ **Fonction HEX2INT()**

Cette fonction transforme le contenu d'un tableau qui est en HEXA en sa valeur INTEGER .

▣ **Fonction INT2HEX()**

Cette fonction transforme le contenu d'un tableau qui est en INTEGER en sa valeur HEXA

▣ **Fonction ASCII2HEXA()**

Cette fonction transforme le contenu d'un tableau qui est en ASCII en sa valeur HEXA .

LIBRAIRIE GRAPHIQUE .H

▣ **Fonction detection()**

Cette fonction sert à initialiser le mode graphique en détectant le "driver" .

▣ **Fonction fenetre()**

Cette fonction sert à tracer une fenêtre en indiquant les coordonnées du coin haut gauche, puis du coin haut droit, puis du coin bas gauche, puis du coin bas droit et en indiquant également la couleur de font .

▣ **Fonction Affiche_G()**

Cette fonction affiche à l'écran un texte à des coordonnées distinctes et avec une couleur définie .

▣ **Fonction Initecran()**

Cette fonction efface l'écran et ensuite affiche les bonnes fenêtres .

▣ **Fonction SYNCHRO_ECRAN()**

Cette fonction écrite en Assembleur sert à synchroniser l'affichage de l'heure avec le balayage du faisceau écran afin d'éliminer les scintillements .

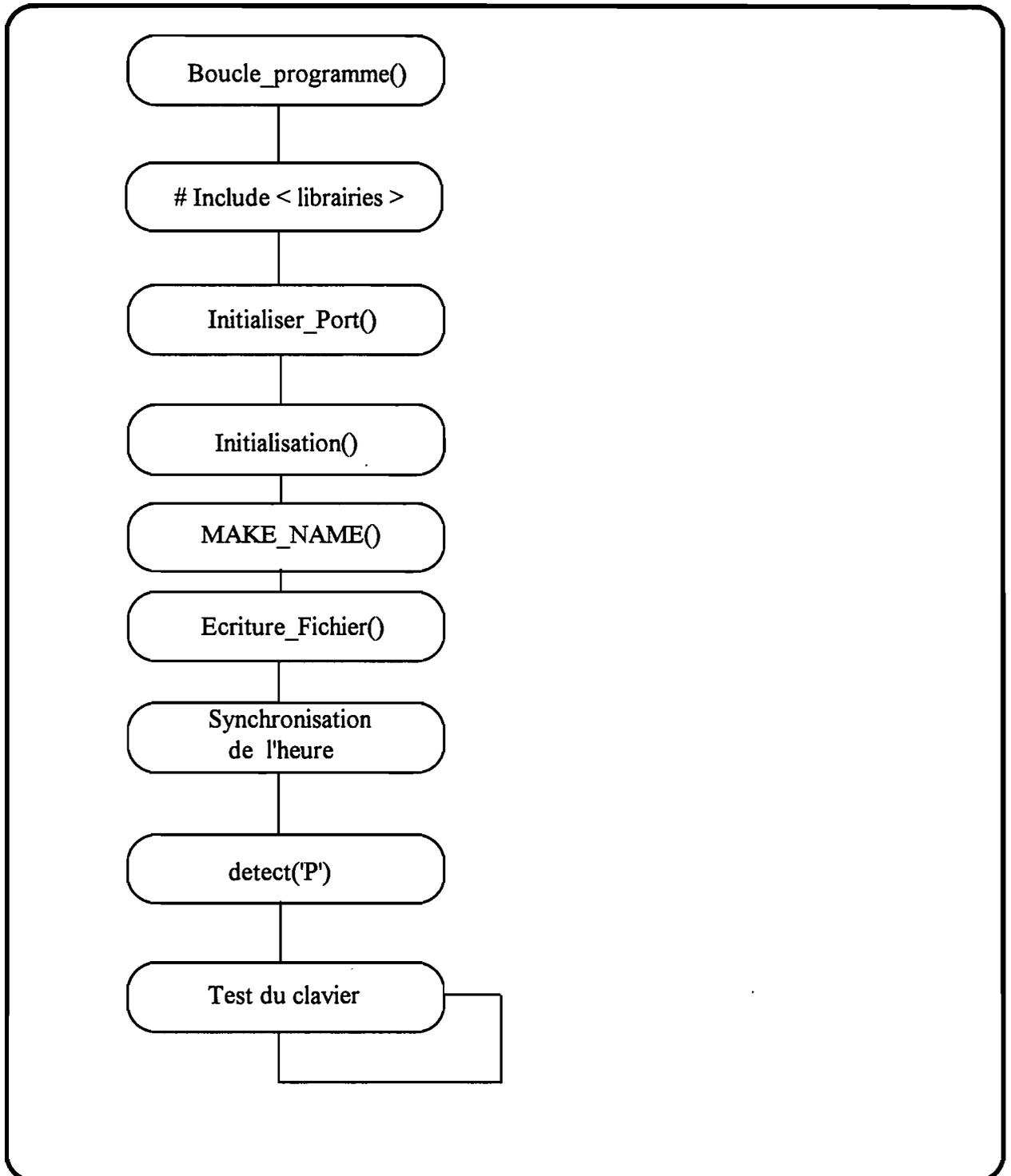
III/6 GRAPHSETS

III/6.1 Introduction

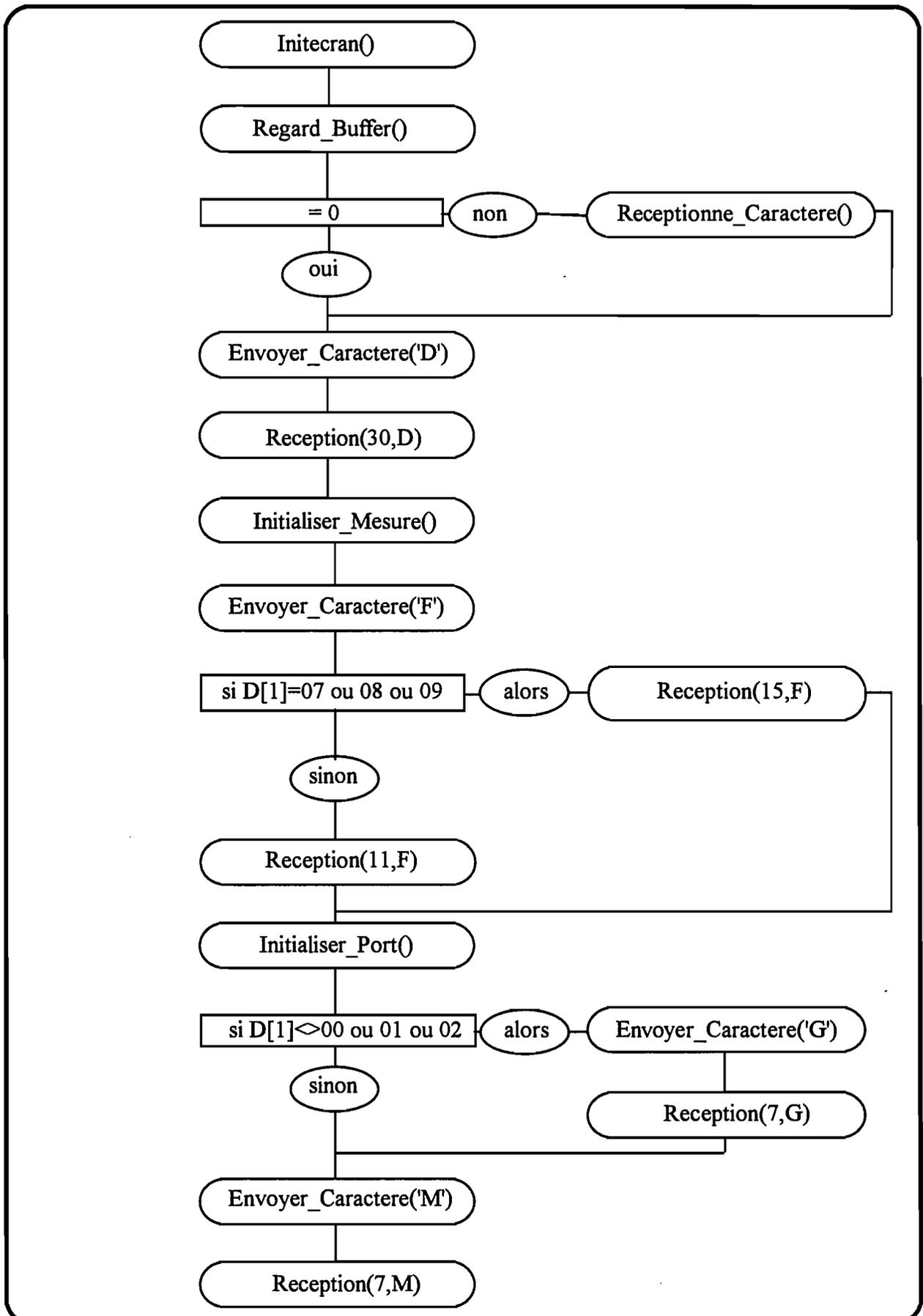
Nous allons développer dans cette partie les graphsets des fonctions qui nous semblent les plus importantes .

- A) Fonction Boucle_programme()
- B) Fonction Initialisation()
- C) Fonction Detect()

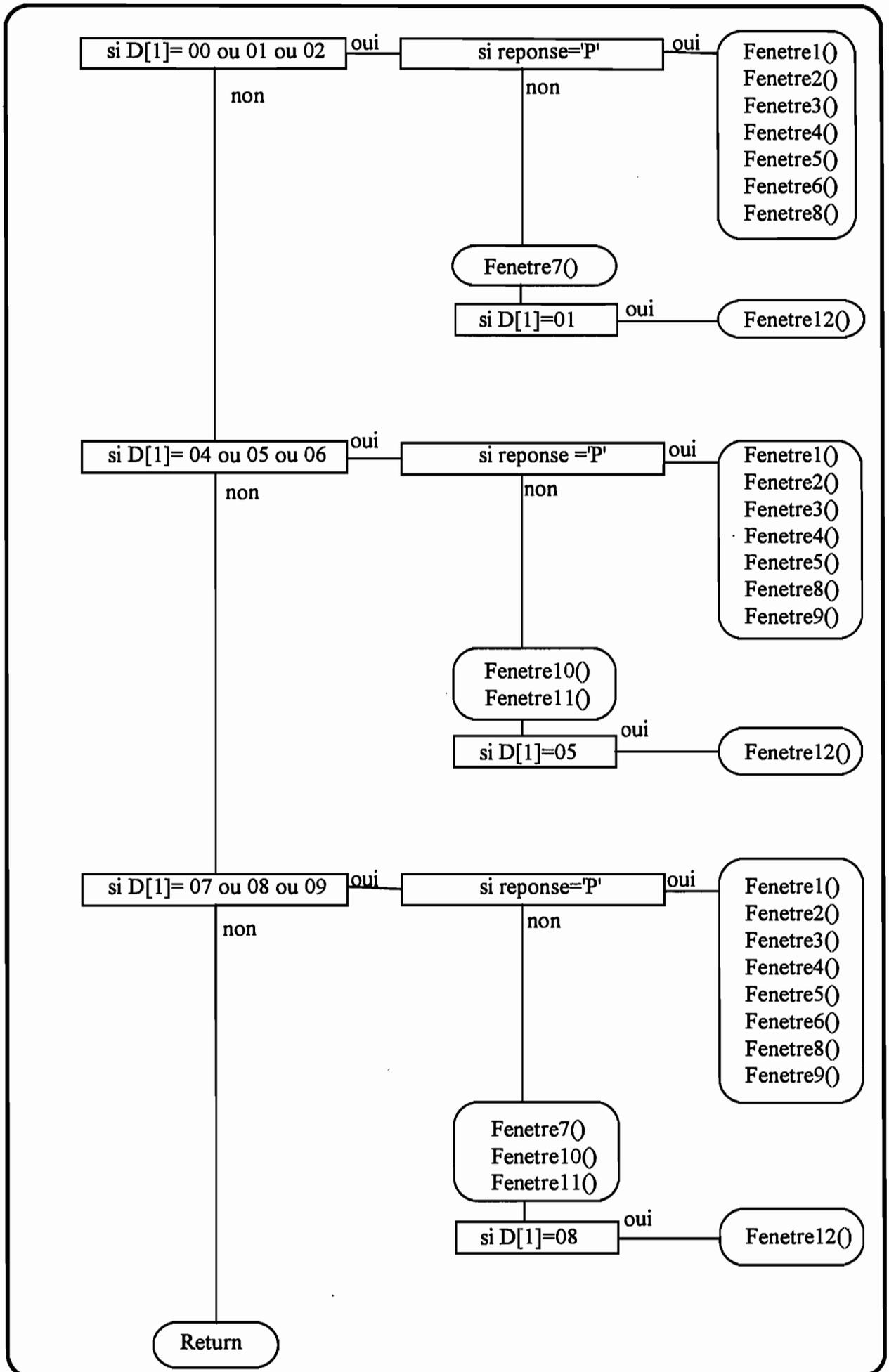
A) Fonction Boucle_programme()



B) Fonction Initialisation()



C) Fonction detect(char reponse)



CHAPITRE IV
PROGRAMMATION

CHAPITRE IV PROGRAMMATION

IV/1 INTRODUCTION

Dans cette partie, nous allons fournir les listings des différentes bibliothèques qui ont été développées.

Ces bibliothèques sont au nombre de 5 et ont les noms suivants :

| | |
|-----------|--------------|
| Librairie | SERIAL .H |
| Librairie | GRAPHIQUE .H |
| Librairie | DISK .H |
| Librairie | CONVERT .H |
| Librairie | TEXTE .H |

Il y aura également le listing du programme principal :

TRM1 .CPP

La signification et l'utilité des fonctions ont été définies dans le chapitre III, ainsi que dans les 3 principaux graphsets .

LIBRAIRIE

SERIAL .H

```

/* NOM : SERIAL.H */
/* Librairie contenant diverses fonctions s rieales */

```

```

/*


RS232.H



Ce programme utilise le Port S rie : 1



RAPPEL DES REGISTRES



|     |                                                |       |
|-----|------------------------------------------------|-------|
|     | Buffer de R ception.....                       | 0x3F8 |
|     | Buffer d' mission.....                         | 0x3F8 |
| il, | Octet Faible du diviseur(DLAB).....            | 0x3F8 |
|     | Octet Fort du diviseur(DLAB).....              | 0x3F9 |
|     | Registre Validation des Interruptions.....     | 0x3F9 |
|     | Registre d'identification des Interruptions... | 0x3FA |
|     | Registre de Contr le de la Ligne.....          | 0x3FB |
|     | Registre de Contr le du Modem.....             | 0x3FC |
|     | Registre d' tat de Ligne.....                  | 0x3FD |
|     | Registre d' tat du modem.....                  | 0x3FE |



Si vous d sirez travailler sur un autre port s rie, remplacez les adresses du port 1, par celles attribu es   votre choix.


*/

```

```

/*-----*/
#define OUI 0x01 /* Variable utilis e pour quitter le programme */
#define INTERRUPT_SCR int86(0x10,&inregs,&outregs);
#define SETTINGS (0xC0 | 0x00 | 0x00 | 0x03 )
#define SETTINGS1 (0xC0 | 0x18 | 0x00 | 0x02 )
#define COM1 0
#define DATA_READY 0x100

```

```

/*-----*/
/*


Fichiers   inclure


*/

```

```

#include "stdio.h"
#include "stdlib.h"
#include "dos.h"
#include "fcntl.h"
#include "conio.h"
#include "io.h"
#include "bios.h"
#include "sys/stat.h"
#include <math.h>
#include "convert.h"
#include "disk.h"

```

```

union REGS inregs,outregs;

unsigned long H[30],I[7],D[30],M[7],G[7],F[17],L[4];
/*
TABLEAUX DE RECEPTION */

float F1[15]; /* TABLEAU POUR LES EEPROMS */
int test=0; /* BOOLEEN pour changement d' cran */
char c = 0x00; /* CARACTERE RECU */
int compteur_og=0; /* NOMBRE D'AUGET POUR LE TEST */
int index=1; /* INDEX POUR LES TABLEAUX */
float tampon; /* Pour l'Affichage des Nombres a Virgule */
char Name[12]; /* NOM DU FICHIER */
/*-----*/

```

```

/* sont vides alors ok. */
}
/*-----*/
/*


ENVOYER UN CARACTERE


*/
void Envoyer_Caractere(unsigned char caractere) /* envoie un caractère sur la ligne
{
    outportb(0x3F8,caractere); /* Buffer d'Emission, caractère */
}
/*-----*/
/*


REGARDE DANS LE BUFFER DE RECEPTION LES CARACTERES RECUS


*/
Regard_Buffer()
{
    char resultat; /* déclaration de la variable resultat */

    resultat = inportb(0x3FD); /* inspecte le Registre d'Etat de Ligne */
    if (resultat >=0x7F) /* si resultat = 1111111 */
    {
        printf(t6);
        printf("\n");
        return(-1); /* la fonction retourne 1 (1 = erreur) */
    }
    if (resultat & 0x10) /* si resultat = 10000 */
    {
        printf(t7);
        printf("\n");
        return(-1); /* la fonction retourne 1 (1 = erreur) */
    }
    if (resultat & 0x08) /* si resultat = 1000 */
    {
        printf(t8);
        printf("\n");
        return(-1); /* la fonction retourne 1 (1 = erreur) */
    }
    if (resultat & 0x01) /* si resultat = 1. caractère disponible */
        return(0); /* la fonction retourne alors 0 (0 = OK) */
    else
        return(-1);
}
/*-----*/
/*


RECEPTIONNE UN CARACTERE


*/
/* ENTREE : TABLEAU */
/* SORTIE : TABLEAU CHARGE DU TEMPON */

void Receptionne_Caractere()
{
    c=inportb(0x3F8);
}
/*-----*/
/*


RECEPTIONNE DANS UN TABLEAU HISTO


*/

```

```

/*


ENVOYER UN CARACTERE SUR L'ECRAN AVEC DEPLACEMENT DU CURSEUR


*/

```

```
void Scr_Echo(unsigned char caractere)
```

```

{
    inregs.h.ah = 0x0E;          /* Ecriture d'un caractere sur l'écran avec */
                                /* avance du curseur.                       */
    inregs.h.al = caractere;    /* al contient le caractère à écrire       */
    inregs.h.bl = 0x07;        /* bl contient la couleur du caractère     */
    inregs.h.bh = 00;          /* bh contient le numéro de la page       */
    INTERRUP_SCR
}

```

```

/*


TRAITEMENT DE LA RS232


Initialiser_Mesure()
Initialiser_Port()
Ready()
Envoyer_Caractere()
Regard_Buffer()
Receptionne_Caractere()
Reception_H()
Reception()
*/

```

```

/*


INITIALISE LE PORT DE COMMUNICATION  

    Vitesse = 4800 : 7 bits Data : Parité Paire : 1 bit de Stop


*/

```

```

void Initialiser_Mesure()
{
    bioscom (0, SETTINGS1, COM1);
}

```

```

/*


INITIALISE LE PORT DE COMMUNICATION  

    Vitesse = 4800 : 8 bits Data : : 1 bit de Stop


*/

```

```

void Initialiser_Port()
{
    bioscom (0, SETTINGS, COM1);
}

```

```
/*-----*/
```

```

/*


BOUCLE D'ATTENTE AVANT EMISSION DE CARACTERE


*/

```

```

void Ready()
{
    int resultat;          /* déclaration de la variable resultat */
    do
    {
        resultat = inportb(0x3FD); /* inspecte le Registre d'état de Ligne */
    }
    while (resultat != 0x60); /* 0 1 1 0 0 0 0 0 */
                                /* Pas de bousculade !!! si les buffers */
}

```

```

/* ENTREE : TABLEAU                                     */
/* SORTIE : TABLEAU CHARGE DU TEMPON                   */

```

```

void Reception_H(unsigned long tableau[])
{
int compteur=1;

do
{
if (Regard_Buffer()==0)
{
tableau[compteur]=inport(0x3F8);
compteur++;
}
}
while (tableau[compteur-1]!=48);
}

```

```

/*-----*/

```

```

/*


RECEPTIONNE DANS UN TABLEAU


*/
/* ENTREE : TABLEAU ET NOMBRE DE CARACTERES A RECEVOIR */
/* SORTIE : TABLEAU CHARGE DU TEMPON                   */

```

```

void Reception(int index,unsigned long tableau[])
{
int compteur=1;

do
{
if (Regard_Buffer()==0)
{
tableau[compteur]=inport(0x3F8);
compteur++;
}
}
while (compteur<index);
}

```

```

/*-----*/
/*-----*/

```

```

/*


INITIALISE LES TABLEUX D,F,M,G


*/

```

```

void Initialisation(void)

{
Initecran();
gotoxy(3,5);printf(transf);
if (Regard_Buffer()==0) Receptionne_Caractere();

Envoyer_Caractere('D'); /* Initialise le T.INIT */
delay(10);
Reception(30,D);
gotoxy(25,5);printf(t25);

Initialiser_Mesure();
Envoyer_Caractere('F'); /* Initialise le T.EEPROMS */
delay(1);

if (D[1]==(0x07 || 0x08 || 0x09)) Reception(15,F);

```

```

else Reception(11,F);

Initialiser_Port();
gotoxy(25,5);printf(t50);

if ((D[1]!=0x00) && (D[1]!=0x01) && (D[1]!=0x02))
{
Envoyer_Caractere('G');          /* Initialise le T.G      */
delay(1);
Reception(7,G);
gotoxy(25,5);printf(t75);
}

Envoyer_Caractere('M');          /* Initialise le T.M      */
delay(1);
Reception(7,M);
gotoxy(25,5);printf(t100);

delay(1000);
}

/*-----*/
/*-----*/

/*


FONCTIONS DE GESTION DES FENETRES


*/

FONCTION : FENETRE1
TRAITEMENT DU TYPE DE CENTRALE                                     */

void Fenetre1(void)
{
if ((D[1]==0x00) || (D[1]==0x01) || (D[1]==0x02) ) {
gotoxy(6,5); printf(e211); /* Systeme PLUVIO      */
gotoxy(6,6); printf(e212);
gotoxy(13,6); printf("%d",D[2]*256+D[3]);
if (D[1]==0x00){ gotoxy(6,7); printf(e214);} /* TRANSMISSION NON ARGOS */
if (D[1]==0x01){ gotoxy(6,7); printf(e213);} /* TRANSMISSION ARGOS     */
if (D[1]==0x02){ gotoxy(6,7); printf(e215);} /* TRANSMISSION MODEM    */
}

if ((D[1]==0x04) || (D[1]==0x05) || (D[1]==0x06)) {
gotoxy(6,5); printf(e311);
gotoxy(6,6); printf(e212); /* Systeme LIMNI      */
gotoxy(13,6); printf("%d",D[2]*256+D[3]);
if (D[1]==0x04){ gotoxy(6,7); printf(e214);} /* TRANSMISSION NON ARGOS */
if (D[1]==0x05){ gotoxy(6,7); printf(e213);} /* TRANSMISSION ARGOS     */
if (D[1]==0x06){ gotoxy(6,7); printf(e215);} /* TRANSMISSION MODEM    */
}

if ((D[1]==0x07) || (D[1]==0x08) || (D[1]==0x09)) {
gotoxy(6,5); printf(e411); /* Systeme PLUVIO-LIMNI */
gotoxy(6,6); printf(e212);
gotoxy(13,6); printf("%d",D[2]*256+D[3]);
if (D[1]==0x07){ gotoxy(6,7); printf(e214);} /* TRANSMISSION NON ARGOS */
if (D[1]==0x08){ gotoxy(6,7); printf(e213);} /* TRANSMISSION ARGOS     */
if (D[1]==0x09){ gotoxy(6,7); printf(e215);} /* TRANSMISSION MODEM    */
}
}

/*-----*/
/* FONCTION FENETRE2                                     */

```

```
/* TRAITEMENT DE L'HEURE ET DE LA DATE
```

```
*/
```

```
void Fenetre2(void)
```

```
{
gotoxy(6,9); printf(e6l2);
pendule();
gotoxy(6,10); printf(e6l1);
gotoxy(16,10); printf("%x",D[10]);
gotoxy(18,10);printf(":");
if (D[9]<10) { gotoxy(19,10);printf("0" "%x",D[9]);}
else      { gotoxy(19,10);printf("%x",D[9]); }
gotoxy(21,10);printf(":");
gotoxy(22,10); printf("%x",D[8]);
}      i i,
```

```
/*-----*/
/* FONCTION FENETRE3                               */
/* TRAITEMENT DE L'ETAT DES TENSIONS                */
```

```
void Fenetre3(void)
```

```
{
gotoxy(40,5);printf(e10l1);
if ((tampon=D[25]/10)>=10){ gotoxy(48,5);printf("%.0f",tampon=D[25]/10);}
else      { gotoxy(49,5);printf("%.0f",tampon=D[25]/10);}
gotoxy(50,5);printf("." "%0f",tampon=D[25]%10);

gotoxy(40,6);printf(e10l2);
if ((tampon=D[26]/10)>=10){ gotoxy(48,6);printf("%.0f",tampon=D[26]/10);}
else      { gotoxy(49,6);printf("%.0f",tampon=D[26]/10);}
gotoxy(50,6);printf("." "%0f",tampon=D[26]%10);
}
```

```
/*-----*/
/* FONCTION FENETRE4                               */
/* TRAITEMENT DE LA DUREE DE FONCTIONNEMENT        */
```

```
void Fenetre4(void)
```

```
{
gotoxy(40,9);printf(e8l1);
gotoxy(40,10);printf("%x",D[20]);
gotoxy(42,10);printf(":");
if (D[19]<10) { gotoxy(43,10);printf("0" "%x",D[19]);}
else      { gotoxy(43,10);printf("%x",D[19]); }
gotoxy(45,10);printf(":");
gotoxy(46,10);printf("%x",D[18]);
gotoxy(53,10);printf("Soit");
gotoxy(40,11);printf("%d",D[21]*256+D[22]);
gotoxy(46,11);printf(e8l2);

}
```

```
/*-----*/
/* FONCTION FENETRES                               */
/* TRAITEMENT DE LA TAILLE DE L'AUGET              */
```

```
void Fenetre5(void)
```

```
{
gotoxy(56,5);printf(apor);
if (D[27]==0x0A) { gotoxy(56,6);printf("1");}
else {
gotoxy(56,6);printf("0.");
gotoxy(58,6);printf("%d",D[27]);
}
```

```

    }
}

/*-----*/
/* FONCTION FENETRE6                                     */
/* TRAITEMENT DERNIER BASCULEMENT ET CUMUL BASCULEMENTS */

void Fenetre6(void)
{
gotoxy(6,12);printf(e813);
gotoxy(6,13);printf("%x",D[15]);
gotoxy(8,13);printf(":");
if (D[14]<10) { gotoxy(9,13);printf("0" "%x",D[14]);}
else          { gotoxy(9,13);printf("%x",D[14]); }
gotoxy(12,13);printf("à");
gotoxy(14,13);printf("%x",D[16]);
gotoxy(16,13);printf("h");
if (D[17]<10) { gotoxy(17,13);printf("0" "%x",D[17]);}
else          { gotoxy(17,13);printf("%x",D[17]); }
gotoxy(19,13);printf("mn");

gotoxy(6,14);printf(e814);
gotoxy(6,15);printf("%d",D[4]*256+D[5]);
gotoxy(13,15);printf("Basculements");
}

/*-----*/
/* FONCTION FENETRE7                                     */
/* ETAT DES EEPROMS PARTITION PLUVIO                     */

void Fenetre7(void)
{
ASCII2HEXA(12,F);      /* MISE EN FORME DU TABLEAU "F" */
INTFLOAT(12,F,F1);
gotoxy(40,5);printf(e9);
gotoxy(40,6);printf(part1);

if (D[29]!= 0x04 && D[29]!=0x05 && D[29]!=0x06
    && D[29]!=0x0C && D[29]!=0x0D && D[29]!=0x0E)
{
if (D[1]==0x00 || D[1]==0x01 ||D[1]==0x02) {

gotoxy(40,8);printf("%.0f", (F1[4]+F1[3]*16+F1[2]*256+F1[1]*4096)+1);
gotoxy(48,8);printf(e911);
gotoxy(40,9);printf("%.0f", (F1[8]+F1[7]*16+F1[6]*256+F1[5]*4096));
gotoxy(48,9);printf(e912);

}

if (D[1]==0x07 || D[1]==0x08 ||D[1]==0x09) {

gotoxy(40,8);printf("%.0f", (F1[4]+F1[3]*16+F1[2]*256+F1[1]*4096)-32767);
gotoxy(48,8);printf(e911);
gotoxy(40,9);printf("%.0f", F1[12]+F1[11]*16+F1[10]*256+F1[9]*4096);
gotoxy(48,9);printf(e912);

}

}
else
{
gotoxy(40,7);printf(e91);
gotoxy(40,8);printf(e92);
}
}

```

```

}

/*-----*/
/* FONCTION FENETRES */
/* TRAITEMENT DU TEST DE L'AUGET */

void Fenetre8(void)
{
char reponse;

gotoxy(6,18);printf(e1311);
gotoxy(6,19);printf("%d",compteur_og);
gotoxy(10,19);printf(e1312);

}

/*-----*/
/* FONCTION FENETRE9 */
/* TRAITEMENT DU FORMATAGE */

void Fenetre9(void)
{

gotoxy(40,13);printf(e1514); /* Seuil */
gotoxy(40,14);printf("%d",G[1]);

gotoxy(40,15);printf(e1512); /* Decalage */
gotoxy(40,16);printf("%d",G[4]*256+G[3]);

gotoxy(40,17);printf(e1515); /* Periode */
gotoxy(40,18);printf("%d",G[2]);

gotoxy(40,19);printf(e1513); /* N° SPI */
if (M[6]<10) { gotoxy(41,20);printf("%x",M[6]);}
else { gotoxy(40,20);printf("%x",M[6]);}
if (M[5]<10) { gotoxy(42,20);printf("0" "%x",M[5]);}
else { gotoxy(42,20);printf("%x",M[5]); }
}

/*-----*/
/* FONCTION FENETRE 10 */
/* ETAT DES EEPROMS PARTITION LIMNI */

void Fenetre10(void)
{
gotoxy(40,11);printf(e9);
gotoxy(40,12);printf(part2);

ASCII2HEXA(12,F); /* MISE EN FORME DU TABLEAU "F" */
INTFLOAT(12,F,F1);

if ((D[29]!= 0x04) && (D[29]!=0x05) && (D[29]!=0x06)
&& (D[29]!=0x0C) && (D[29]!=0x0D) && (D[29]!=0x0E))
{
if ((D[1]==0x04) || (D[1]==0x05) || (D[1]==0x06)) {

gotoxy(40,14);printf("%.0f", (F1[4]+F1[3]*16+F1[2]*256+F1[1]*4096)+1);
gotoxy(48,14);printf(e911);
gotoxy(40,15);printf("%.0f", F1[8]+F1[7]*16+F1[6]*256+F1[5]*4096);
gotoxy(48,15);printf(e912);

}

if ((D[1]==0x07) || (D[1]==0x08) || (D[1]==0x09)) {

```



```
void detect(char reponse)
{
    if ((D[1]==0x00) || (D[1]==0x01) || (D[1]==0x02))    /* CENTRALE PLUVIO */
        {
            if (reponse=='P')
                {
                    test=0;
                    Initecran();
                    Affiche_G(20,330,mess3,4);
                    Fenetre1();
                    Fenetre2();
                    Fenetre3();
                    Fenetre4();
                    Fenetre5();
                    Fenetre6();
                    Fenetre8();
                }

            if (reponse=='N')
                {
                    Initecran();
                    Affiche_G(20,320,mess2,4);
                    Fenetre7();
                    if (D[1]==0x01) {
                        Fenetre12();
                    }
                }
        }

    if (D[1]==0x04 || D[1]==0x05 || D[1]==0x06)    /* CENTRALE LIMNI */
        {
            if (reponse=='P')
                {
                    test=0;
                    Initecran();
                    Affiche_G(20,330,mess3,4);
                    Fenetre1();
                    Fenetre2();
                    Fenetre3();
                    Fenetre4();
                    Fenetre5();
                    Fenetre8();
                    Fenetre9();
                }
            if(reponse=='N')
                {
                    Initecran();
                    Affiche_G(20,320,mess2,4);
                    if (D[1]==0x05){
                        Fenetre12();
                    }

                    Fenetre10();
                    Fenetre11();
                }
        }

    if (D[1]==0x07 || D[1]==0x08 || D[1]==0x09)    /* CENTRALE PLUVIO-LIMNI */
        {
            if (reponse=='P')
                {
                    test=0;
```

```

    Initecran();
    Affiche_G(20,330,mess3,4);
    Fenetre1();
    Fenetre2();
    Fenetre3();
    Fenetre4();
    Fenetre5();
    Fenetre6();
    Fenetre8();
    Fenetre9();
  }
  if (reponse=='N')
  {
    Initecran();
    Affiche_G(20,320,mess2,4);
    Fenetre7();
    Fenetre10();
    Fenetre11();
    if (D[1]==0x08) Fenetre12();
  }
}

```

```

/*


SAUVEGARDE DES TABLEAUX SUR DISQUE


*/

```

```

/* FONCTION: CREER LE NOM DU FICHIER
/* DETECTE LE TYPE DE CENTRALE
/* DETECTE LE NUMERO DE LA CENTRALE
/* DESTINA=NUMERO.TYPE
*/

```

```

void MAKE_NAME(void)
{
  int *chaine1[8];
  char *chaine2[4];

  ultoa(D[2]*256+D[3],Name,10);

  if ((D[1]==0x00) || (D[1]==0x01) || (D[1]==0x02)) *chaine2=".PLU";
  if (D[1]==0x04 || D[1]==0x05 || D[1]==0x06) *chaine2=".LIM";
  if (D[1]==0x07 || D[1]==0x08 || D[1]==0x09) *chaine2=".P-L";
  strcat(Name,*chaine2);
}

```

```

/*


RECEPTIONNE ET AFFICHE LES CUMULS
    SOUS LA FORME D'UN HISTOGRAMME


*/

```

```

void Histogramme(int n)
{
  Reception(n,I);
  if (I[n-5]==48) { I[n-3]=I[n-4]=I[n-5]=I[n-2]=I[n-1]=0; }
  gotoxy(3,8); printf(e611);
  if (I[n-3]<10) { gotoxy(16,8);printf("0" "%x",I[n-3]);}
  else { gotoxy(16,8);printf("%x",I[n-3]); }
  gotoxy(18,8);printf(":");
  if (I[n-4]<10) { gotoxy(19,8);printf("0" "%x",I[n-4]);}
  else { gotoxy(19,8);printf("%x",I[n-4]); }
  gotoxy(21,8);printf(":");
  if (I[n-5]<10) { gotoxy(22,8);printf("0" "%x",I[n-5]);}
  else { gotoxy(22,8);printf("%x",I[n-5]); }
}

```

```

gotoxy(3,9);printf(e814);
gotoxy(16,9);printf("%d      ",(I[n-2]*256+I[n-1])*D[27]);
gotoxy(20,9);printf("  (1/10 mm)");
gotoxy(3,10);printf(eff);
}

```

```

/*


FONCTION BOUCLE_PROGRAMME



TRAITEMENT DES TABLEAUX DE RECEPTION EN FONCTION DES CENTRALES



ENTREE: TABLEAUX D, F, M, G



SORTIE : VISUELLE


*/

```

```

void Boucle_programme(void)
{
char reponse;          /* déclaration de la variable "reponse" */
char rep;
unsigned char taille;
int temps;
int seuil;
int periode;          /* TAILLE POUR INIT */
int decalage;
unsigned int annee,mois,jour,heure,minute,seconde; /* INIT PENDULES */
unsigned int Sjour,Smois,Sannee; /* Pour histogramme */
int k,l;              /* index sur H */
int x;                /* Affichage sur histogramme en abscisse */

Initialiser_Port();
delay(10);
Initialisation();

MAKE_NAME();
Ecriture_Fichier(D,F,M,G,29,12,6,6,Name); /* SAUVEGARDE DE L'INIT */

```

```

/*-----*/
/*          REGLAGE DE L'HORLOGE PC AVEC CELLE DE LA CENTRALE          */
/*-----*/

```

```

HEX2INT(D,11,L,1);
HEX2INT(D,12,L,2);
HEX2INT(D,13,L,3);

```

```

t.ti_hour=L[1];
t.ti_min=L[2];
t.ti_sec=L[3];
settime(&t);

```

```

/*-----*/
/*          AFFICHAGE DE LA 1ère PAGE DE FENETRES          */
/*-----*/

```

```

detect('P');

```

```

/*


BOUCLE DE TEST DE FRAPPE AU CLAVIER



LANCE DES PROCEDURES SELON LE CHOIX EFFECTUE :


*/

```

```

(Q) QUITTER
(N) NEXT
(P) PRECEDENT
(S) MESURE DU SPI
(E) EFFACEMENT DES EEPROMS
(A) AUGET TAILLE
(M) MESSAGE ARGOS
(D) DATE ET HEURE
(H) HISTOGRAMME
(T) TEST AUGET
(F) FORMATAGE EEPROMS

```

```
*/
```

```

do      ii,
{
if (kbhit()) reponse=toupper(getch());
/*  Teste une éventuelle frappe au clavier

```

```
*/
```

```

/*-----*/
/*                      HISTOGRAMME                      */
/*-----*/

```

```

if ((reponse=='H') && ((D[1]==0x00) || (D[1]==0x01) || (D[1]==0x02)
|| (D[1]==0x07) || (D[1]==0x08) || (D[1]==0x09)))
{
if (D[29]>=0x08)
{
Initecran();
gotoxy(3,8);printf(e16l5);
delay(3000);
}
else
{
Initecran();
Affiche_G(470,310,mess9,4);

if ((D[29]==0x04) || (D[29]==0x05) || (D[29]==0x06) || (D[29]==0x0c)
|| (D[29]==0x0D) || (D[29]==0x0E))
{
gotoxy(3,5);printf(e12l2);
delay(2000);
detect('P');
reponse=' ';
}
else
{
gotoxy(3,6);printf(tjour);gotoxy(8,6);fscanf(stdin,"%x",&Sjour);
gotoxy(14,6);printf(tmois);gotoxy(19,6);fscanf(stdin,"%x",&Smois);
gotoxy(27,6);printf(tannee);gotoxy(33,6);fscanf(stdin,"%x",&Sannee);

if ((Sjour>49) || (Smois>12))
{
gotoxy(3,11);printf(e5l3);
delay(2000);
}
else
{
if ((D[4]==0) && (D[5]==0)) {
gotoxy(3,7);printf(e12l5);
delay(2000);
detect('P');
reponse=' ';
}
}
}
}
}

```



```

        if (l>9) { x=40; l=0; } /* TESTE LE DEBORDEMENT ECRAN EN Y */
    }
    while (H[k+1]!=48);
    rep=' ';
}
while(rep!='Q');
}

Envoyer_Caractere('D'); /* INIT BUFFER RECEPTION */
delay(10);
}
}
rep=' ';
reponse=' ';
detect('P');
}
}

/*-----*/
/*                          TEST AUGET                          */
/*-----*/

if (reponse=='T')
{
detect('P');
Affiche_G(470,320,mess10,4);
Envoyer_Caractere('B');
delay(10);

do
{
do
{
Receptionne_Caractere();
if (kbhit()) {
c='O';
compteur_og--;
}
}
while(c!='O');

compteur_og++;
gotoxy(6,19);printf("%d",compteur_og);
Envoyer_Caractere('D'); /* Vide le BUFFER */
delay(500);
}
while (!kbhit());

Envoyer_Caractere('Z');
delay(10);
Envoyer_Caractere('D'); /* Vide le BUFFER */
Affiche_G(470,320,mess10,15);
reponse=' ';
detect('P');
}

/*-----*/
/*                          EFFACEMENT DES EEPROMS                          */
/*-----*/

if (reponse=='E')
{
Initecran();
Affiche_G(250,310,mess5,4);
gotoxy(3,5);printf(e1611);
}

```

```

gotoxy(3,6);printf(e1614);
        if (D[29]>=0X08)
            {
                gotoxy(3,8);printf(e1615);
                delay(3000);
                Initecran();
            }
        else
            {
                gotoxy(3,8);printf(e1612);
                rep=getche();
                if (rep=='O')
                    {
                        gotoxy(3,9);printf(e1616);
                        Envoyer_Caractere('E');
                        delay(10);

                        do {
                            Receptionne_Caractere();
                            if (c=='N') break;
                        }
                        while(c!='O');

                        if (c=='O') {
                            gotoxy(3,10);
                            printf(e1617);
                            delay(1000);
                            Envoyer_Caractere('D');
                            delay(10);
                            gotoxy(3,11);printf(e1619);
                            Envoyer_Caractere('R');
                            delay(3000);
                        }
                        else
                            {
                                gotoxy(3,10);
                                printf(e1618);
                                delay(1000);
                                Envoyer_Caractere('D');
                            }
                    }
            }

        reponse=' ';
        detect('P');
    }

/*-----*/
/*          MESURE DU SPI          */
/*-----*/

if ((reponse=='S') && ((D[1]==0x04) || (D[1]==0x05) || (D[1]==0x06)
|| (D[1]==0x07) || (D[1]==0x08) || (D[1]==0x09)))
    {
        if (D[29]>=8)
            {
                Initecran();
                gotoxy(3,8);printf(e1615);
                delay(3000);
            }
        else
            {
                test=1; detect('N');
                Affiche_G(20,340,mess4,4);
                gotoxy(6,7);printf(e1412);
                gotoxy(6,8);printf("?????");
                gotoxy(6,9);printf(e1413);
                gotoxy(6,10);printf("?????");
            }
    }

```

```

        if (Regard_Buffer()==0) Receptionne_Caractere();
        Envoyer_Caractere('M');
        delay(1);
        Reception(7,M);
    }
    reponse=' ';
    detect('N');
}

/*-----*/
/*
      MODIFICATION TAILLE AUGET
*/
/*-----*/

if (reponse=='A')
    {
        if (D[29]>=0X08)
            {
                Initecran();
                gotoxy(3,8);printf(e1615);
                delay(3000);
            }
        else
            {
                Initecran();
                Affiche_G(250,320,mess6,4);
                gotoxy(3,5);printf(p1);
                gotoxy(45,5);fscanf(stdin,"%d",&taille);

                if (taille>10) {
                    gotoxy(43,6);printf(e712);
                    delay(2000);
                }

                else
                    {
                        Envoyer_Caractere('Y');
                        delay(10);
                        do
                            Receptionne_Caractere();
                        while (c!='Y');

                        Envoyer_Caractere(taille);
                        do
                            Receptionne_Caractere();
                        while (c!='O');
                        D[27]=taille;
                        Envoyer_Caractere('D');      /* Vide le BUFFER      */
                    }
                }
            reponse=' ';
            detect('P');
        }

/*-----*/
/*
      MODIFICATION MESSAGE ARGOS
*/
/*-----*/

if ((reponse=='M') && ((D[1]==0x01) || (D[1]==0x05) || (D[1]==0x08)))
    {
        if (D[29]>=0X08)
            {
                Initecran();
                gotoxy(3,8);printf(e1615);
                delay(3000);
            }
        else
            {

```

```

Initecran();
Affiche_G(250,330,mess7,4);
gotoxy(3,5);printf(e1111);
gotoxy(3,6);printf(e1112);
gotoxy(3,7);printf(e1113);
gotoxy(28,6);fscanf(stdin,"%d",&temps);

if (temps>58) {
    gotoxy(3,7);printf(e712);
    delay(2000);
}

else
{
    Envoyer_Caractere('A');
    delay(10);
    Envoyer_Caractere(temps);
    delay(10);
    do
    {
        Receptionne_Caractere();
        if (c=='N') {
            gotoxy(3,7);printf(e11121);
            delay(2000);
            break;
        }
    }
    while (c!='O');
    D[28]=temps;

    Envoyer_Caractere('R'); /* Initialisation */
    delay(100);

    Envoyer_Caractere('D'); /* Vide le BUFFER */
}
reponse=' ';
detect('N');
test=1;
}

/*-----*/
/*          FORMATAGE EEPROMS          */
/*-----*/

if ((reponse=='F') && ((D[1]==0x04) || (D[1]==0x05) || (D[1]==0x06)
|| (D[1]==0x07) || (D[1]==0x08) || (D[1]==0x09)))
{
    if (D[29]>=0x08)
    {
        Initecran();
        gotoxy(3,8);printf(e1615);
        delay(3000);
    }
    else
    {
        Initecran();
        Affiche_G(470,330,mess11,4);
        gotoxy(3,5);printf(e1511);

        gotoxy(3,6);printf(e1514);
        gotoxy(35,6);fscanf(stdin,"%d",&seuil);

        gotoxy(3,7);printf(e1515);
        gotoxy(35,7);fscanf(stdin,"%d",&periode);
    }
}

```

```

gotoxy(3,8);printf(e15l2);
gotoxy(35,8);fscanf(stdin,"%d",&decalage);

if ((seuil>99) || (periode>99) || (decalage>999)
|| (decalage<0))
{ gotoxy(3,9);printf(e7l2);
  delay(2000);
}
else
{
if (abs(decalage)<=255) {
    G[4]=0x00;
    G[3]=abs(decalage);
}

if ((abs(decalage)>255) && (abs(decalage)<=511))
    {
    G[4]=0x01;
    G[3]=abs(decalage)-256;
    }

if ((abs(decalage)>511) && (abs(decalage)<=767))
    {
    G[4]=0x02;
    G[3]=abs(decalage)-512;
    }

if (abs(decalage)>767) {
    G[4]=0x03;
    G[3]=abs(decalage)-768;
    }

G[1]=seuil;
G[2]=periode;

gotoxy(3,9);printf(transf);
Envoyer_Caractere('I');
delay(10);
do
Receptionne_Caractere();
while (c!='I');

Envoyer_Caractere(seuil);
delay(10);
Envoyer_Caractere(periode);
delay(10);
Envoyer_Caractere(G[3]);
delay(10);
Envoyer_Caractere(G[4]);
delay(10);

do
Receptionne_Caractere();
while (c!='O');

Envoyer_Caractere('D');      /* Vide le BUFFER      */
}
}
reponse=' ';
detect('P');
}

```

```

/*-----*/
/*                                MODIFICATION DATE ET HEURE                                */
/*-----*/

```

```

/*-----*/
if (reponse=='D')
{
    if (D[29]>=0X08)
        {
            Initecran();
            gotoxy(3,8);printf(e1615);
            delay(3000);
        }
    else
    {
        Initecran();
        Affiche_G(250,340,mess8,4);

        gotoxy(3,5);printf(theure);gotoxy(12,5);fscanf(stdin,"%x",&heure);
        gotoxy(3,6);printf(tminute);gotoxy(12,6);fscanf(stdin,"%x",&minute);
        gotoxy(3,7);printf(tseconde);gotoxy(12,7);fscanf(stdin,"%x",&seconde);
        gotoxy(3,8);printf(tjour);gotoxy(12,8);fscanf(stdin,"%x",&jour);
        gotoxy(3,9);printf(tmois);gotoxy(12,9);fscanf(stdin,"%x",&mois);
        gotoxy(3,10);printf(tannee);gotoxy(12,10);fscanf(stdin,"%x",&annee);

        if ((heure>=24) || (minute>=60) || (seconde>=60) || (jour>49) || (mois>12))
            {
                gotoxy(3,11);printf(e513);
                delay(2000);
            }
            else
            {
                Envoyer_Caractere('H');
                delay(10);

                do
                    Receptionne_Caractere();
                while(c!='H');

                Envoyer_Caractere(annee);
                delay(10);
                Envoyer_Caractere(mois);
                delay(10);
                Envoyer_Caractere(jour);
                delay(10);
                Envoyer_Caractere(heure);
                delay(10);
                Envoyer_Caractere(minute);
                delay(10);
                Envoyer_Caractere(seconde);
                delay(10);

                do
                {
                    Receptionne_Caractere();

                    if (c=='N')
                    {
                        gotoxy(3,11);printf(e512);
                        delay(2000);
                        break;
                    }
                }
                while(c!='O');

                if (c=='O')
                {
                    gotoxy(3,11);printf(e511);
                    delay(2000);
                }
            }
    }
}

```

```

D[8]=annee;
D[9]=mois;
D[10]=jour;
D[11]=heure;
D[12]=minute;
D[13]=seconde;

HEX2INT(D,11,L,1);
HEX2INT(D,12,L,2);
HEX2INT(D,13,L,3);

t.ti_hour=L[1];
t.ti_min=L[2];
t.ti_sec=L[3];
settime(&t);
}

Envoyer_Caractere('D');
}
}
reponse=' ';
detect('P');
}

/*-----*/
/*                INITIALISATION                */
/*-----*/

if (reponse=='I')
{
Initialisation();
Affiche_G(470,340,mess12,4);
reponse=' ';
detect('P');
}

/*-----*/
/*                NEXT                */
/*-----*/

if (reponse=='N')
{
detect('N');
reponse=' ';
test=1;
}

/*-----*/
/*                PRECEDENT                */
/*-----*/

if (reponse=='P')
{
detect('P');
reponse=' ';
test=0;
}

/*-----*/
/*                TEST POUR LA SYNCHRO                */
/*-----*/

```

```
if (test!=1) {
    SYNCHRO_ECRAN();
    pendule();
}

while (reponse!='Q'); /* FIN DE TEST CLAVIER */

closegraph(); /* RETOUR EN MODE TEXTE */
clrscr();
}

/* il, Fin du Programme de */
/*-----*/
```

LIBRAIRIE
GRAPHIQUE .H

```

/* NOM : GRAPHIQUE.H                                                                    */
/*-----*/
/* Librairie contenant diverses fonctions graphiques                                    */
/*-----*/

#include <graphics.h>
#include <process.h>
#include <stdlib.h>
#include <ctype.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>
#include <string.h>

#define VERT_RETRACE 0x3DA

/*-----*/
/* FONCTION DETECTION                                                                    */
/* Recherche et Passage en mode graphic maximum                                        */
/* Retourne message erreur                                                            */
/*-----*/

void detection(void)
{
    int gdriver =9, gmode=VGAMED, errorcode;

    initgraph(&gdriver, &gmode,"c:\\tc3\\bgi");
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("erreur graphique: %s\n", grapherrormsg(errorcode));
        printf("Frappez une touche pour stopper:");
        getch();
        exit(1); // sortie du programme
    }
}

/*-----*/
/* FONCTION FENETRE                                                                    */
/* Tracé d'une fenêtre de couleur col et de dim hgx,hgy,bdx,bdy                    */
/* Sortie visuelle écran                                                            */
/*-----*/

void fenetre(int hgx,int hgy,int bdx, int bdy,int col)
    // pour le tracé des fenêtres
{
    int poly[8];
    int excol=getcolor(); // définition de la couleur

    poly[0]=hgx; // 1
    poly[1]=hgy;
    poly[2]=hgx; // 2
    poly[3]=bdy;
    poly[4]=bdx; // 3
    poly[5]=bdy;
    poly[6]=bdx; // 4
    poly[7]=hgy;
    setfillstyle(SOLID_FILL,col); //motif de remplissage
    setcolor(col);
    fillpoly(4,poly);
    setcolor(excol);
}

/*-----*/

```

```

/* FONCTION AFFICHE_G                                                    */
/* Affiche un texte à l'écran, avec des coordonnées et une couleur donnés */
/* Sortie visuelle écran Graphique                                     */
/*-----*/

void Affiche_G(int x,int y,char texte[60],int color)
{
setcolor(color);
outtextxy(x,y,texte);
setcolor(WHITE);
}

/*-----*/
/* FONCTION INITECRAN                                                    */
/* Réinitialise les fenêtres                                           */
/* Sortie visuelle écran                                               */
/*-----*/

void Initecran(void)
{
cleardevice();
setcolor(BLACK);
fenetre(10,10,630,40,CYAN);
fenetre (10,290,630,350,BLUE);          /* AIDE                               */
outtextxy(200,20,t11);
setcolor(WHITE);
Affiche_G(250,290,mess0,15);
Affiche_G(20,310,mess1,15);
Affiche_G(20,320,mess2,15);
Affiche_G(20,330,mess3,15);
Affiche_G(20,340,mess4,15);
Affiche_G(250,310,mess5,15);
Affiche_G(250,320,mess6,15);
Affiche_G(250,330,mess7,15);
Affiche_G(250,340,mess8,15);
Affiche_G(470,310,mess9,15);
Affiche_G(470,320,mess10,15);
Affiche_G(470,330,mess11,15);
Affiche_G(470,340,mess12,15);
}

/*-----*/
/* FONCTION : SYNCHRO_ECRAN                                             */
/* Attend la synchro horizontale et verticale pour afficher          */
/*-----*/

void SYNCHRO_ECRAN(void)
{
asm    mov    dx,VERT_RETRACE;
sp3;;          /* SYNCHRO VERTICALE                               */
asm          in    al,dx;
asm          test  al,8;
asm          jne   sp3;
sp4;;          /* RETOUR FAISCEAU                               */
asm          in    al,dx;
asm          test  al,8;
asm          je    sp4;
}

/*-----*/

```

LIBRAIRIE
CONVERT .H

```

/* NOM : CONVERT.H */
/* Librairie contenant diverses fonctions de conversion */

/*-----*/
/*          HORLOGE EN TEMPS COMPENSE          */
/*-----*/

struct time t; /* RECUPERATION DONNEES TIME */

void pendule(void)
{
    gettime(&t);
    if (t.ti_hour<10) { gotoxy(16,9);printf("0" "%d",t.ti_hour); }
    else { gotoxy(16,9);printf("%d",t.ti_hour); }

    if (t.ti_min<10) { gotoxy(19,9);printf("0" "%d",t.ti_min); }
    else { gotoxy(19,9);printf("%d",t.ti_min); }

    if (t.ti_sec<10) { gotoxy(22,9);printf("0" "%d",t.ti_sec); }
    else { gotoxy(22,9);printf("%d",t.ti_sec); }

    gotoxy(18,9);printf(":");
    gotoxy(21,9);printf(":");
}

/*


UTILITAIRES DE MISE EN FORME
        DES CARACTERES RECUS


/*-----*/
/*          TRANSFORME UN ENTIER EN NOMBRE FLOTTANT          */
/*-----*/

void INTFLOAT(int nombre,unsigned long entree[],float sortie[])
{
    int i=1;
    for (i=1;i<=nombre;i++) sortie[i]=entree[i];
}

/*-----*/
/*          TRANSFORME UN HEXA EN NOMBRE ENTIER          */
/*-----*/

void HEX2INT(unsigned long tab1[],int i1,unsigned long tab2[],int i2)
{
    if ((tab1[i1]>=0x00) && (tab1[i1]<=0x09)) tab2[i2]=tab1[i1];
    if ((tab1[i1]>=0x10) && (tab1[i1]<=0x19)) tab2[i2]=tab1[i1]-0x06;
    if ((tab1[i1]>=0x20) && (tab1[i1]<=0x29)) tab2[i2]=tab1[i1]-0x0C;
    if ((tab1[i1]>=0x30) && (tab1[i1]<=0x39)) tab2[i2]=tab1[i1]-0x12;
    if ((tab1[i1]>=0x40) && (tab1[i1]<=0x49)) tab2[i2]=tab1[i1]-0x18;
    if ((tab1[i1]>=0x50) && (tab1[i1]<=0x59)) tab2[i2]=tab1[i1]-0x1E;
    if ((tab1[i1]>=0x60) && (tab1[i1]<=0x69)) tab2[i2]=tab1[i1]-0x24;
    if ((tab1[i1]>=0x70) && (tab1[i1]<=0x79)) tab2[i2]=tab1[i1]-0x2A;
    if ((tab1[i1]>=0x80) && (tab1[i1]<=0x89)) tab2[i2]=tab1[i1]-0x30;
    if ((tab1[i1]>=0x90) && (tab1[i1]<=0x99)) tab2[i2]=tab1[i1]-0x36;
}

/*-----*/
/*          TRANSFORME UN ENTIER EN NOMBRE HEXA          */
/*-----*/

void INT2HEX(unsigned long tab1[],int i1,unsigned long tab2[],int i2)
{
    if ((tab1[i1]>=0) && (tab1[i1]<=9)) tab2[i2]=tab1[i1];
}

```

```

if ((tab1[i1]>=10) && (tab1[i1]<=19)) tab2[i2]=tab1[i1]+6;
if ((tab1[i1]>=20) && (tab1[i1]<=29)) tab2[i2]=tab1[i1]+12;
if ((tab1[i1]>=30) && (tab1[i1]<=39)) tab2[i2]=tab1[i1]+18;
if ((tab1[i1]>=40) && (tab1[i1]<=49)) tab2[i2]=tab1[i1]+24;
if ((tab1[i1]>=50) && (tab1[i1]<=59)) tab2[i2]=tab1[i1]+30;
if ((tab1[i1]>=60) && (tab1[i1]<=69)) tab2[i2]=tab1[i1]+36;
if ((tab1[i1]>=70) && (tab1[i1]<=79)) tab2[i2]=tab1[i1]+42;
if ((tab1[i1]>=80) && (tab1[i1]<=89)) tab2[i2]=tab1[i1]+48;
if ((tab1[i1]>=90) && (tab1[i1]<=99)) tab2[i2]=tab1[i1]+54;

```

```

}

```

```

/*-----*/
/*          TRANSFORME DE L' ASCII EN HEXA          */
/*-----*/

```

```

void ASCII2HEXA(int nombre,unsigned long tableau[])

```

```

{
int i=1;
for (i=1;i<=nombre;i++) {
    if (tableau[i]==0x30) tableau[i]=0x00;
    if (tableau[i]==0x31) tableau[i]=0x01;
    if (tableau[i]==0x32) tableau[i]=0x02;
    if (tableau[i]==0x33) tableau[i]=0x03;
    if (tableau[i]==0x34) tableau[i]=0x04;
    if (tableau[i]==0x35) tableau[i]=0x05;
    if (tableau[i]==0x36) tableau[i]=0x06;
    if (tableau[i]==0x37) tableau[i]=0x07;
    if (tableau[i]==0x38) tableau[i]=0x08;
    if (tableau[i]==0x39) tableau[i]=0x09;
    if (tableau[i]==0x41) tableau[i]=0x0A;
    if (tableau[i]==0x42) tableau[i]=0x0B;
    if (tableau[i]==0x43) tableau[i]=0x0C;
    if (tableau[i]==0x44) tableau[i]=0x0D;
    if (tableau[i]==0x45) tableau[i]=0x0E;
    if (tableau[i]==0x46) tableau[i]=0x0F;
}
}

```

LIBRAIRIE

DISK .H

```

/* NOM : DISK.H                                                    */
/* LECTURE - ECRITURE disque                                       */

/* -----*/

#include <sys\stat.h>
#include <string.h>
#include <fcntl.h>
#include <io.h>

/* FONCTION: ECRITURE FICHER                                       */
/* ECRIT DES DONNEES D'UN FICHER DISK                               */
/* ECRIT DANS BUF                                                  */

void Ecriture_Fichier(unsigned long *buf1,unsigned long *buf2,
unsigned long *buf3,unsigned long *buf4,int compt,
int c2,int c3,int c4,char name[])
{
int handle, bytes;

compt=compt*8;
c2=c2*8;
c3=c3*8;
c4=c4*8;

if ((handle = open(name, O_CREAT | O_WRONLY | O_BINARY,
S_IWRITE | S_IREAD)) == -1)
{
printf("Error Opening File\n");
exit(1);
}

if ((bytes = _write(handle, buf1, compt)) == -1)
{
printf("Write Failed.\n");
exit(1);
}

if ((bytes = _write(handle, buf2, c2)) == -1)
{
printf("Write Failed.\n");
exit(1);
}

if ((bytes = _write(handle, buf3, c3)) == -1)
{
printf("Write Failed.\n");
exit(1);
}

if ((bytes = _write(handle, buf4, c4)) == -1)
{
printf("Write Failed.\n");
exit(1);
}
}

```

LIBRAIRIE

TEXTE .H

```

/* NOM : TEXTE.H */
/* LIBRAIRIE CONTENANT LES LIGNES DE TEXTE DU PROGRAMME */

/*-----*/
/*----- PRESENTATION ET RS232 -----*/

char t1[]="EMULATION TERMINAL";
char t11[]="***** EMULATION TERMINAL *****";
char t2[]="ORSTOM";
char t3[]="1993";
char t5[]="CELLULE TECHNOLOGIQUE D'HYDROLOGIE";
char t6[]="TIME OUT";
char t7[]="DETECTION D'UN BREAK";
char t8[]="ERREUR DE BIT START/STOP";
/*-----*/
/*----- ECRAN DU TERMINAL -----*/

char e111[]="VPILE FAIBLE";
char e112[]="EEPROMS HS";
char e113[]="EEPROMS PLEINES";
char e114[]="ENREGISTREMENT BLOQUE !!!!";
char transf[]="TRANSFERT EN COURS...";
char mesure[]="MESURE EN COURS...";
char eff[]="";
char e211[]="SYSTEME PLUVIO";
char e212[]="NUMERO:";
char e213[]="BALISE ARGOS INSTALLEE";
char e214[]="BALISE ARGOS NON INSTALLEE";
char e215[]="TRANSMISSION MODEM INSTALLEE";

char e311[]="SYSTEME LIMNI";

char e411[]="SYSTEME PLUVIO LIMNI";

char e511[]="SYNCHRO OK";
char e512[]="HORLOGE EN PANNE";
char e513[]="DATE INCORRECTE";
char tannee[]="Année:";
char tmois[]="Mois:";
char tjour[]="Jour:";
char theure[]="Heure:";
char tminute[]="Minute:";
char tseconde[]="Seconde:";

char e611[]="DATE : ";
char e612[]="HEURE : ";

char e711[]="ENTRER LA TAILLE DE L'AUGET = ";
char e712[]="MAUVAISE TAILLE !!!!!";
char e811[]="MISE EN ROUTE LE ";
char e812[]="JOURS DE FONCTIONNEMENT";
char e813[]="DERNIER BASCULEMENT AU ";
char e814[]="CUMUL TOTAL";

char e9[]="ETAT DES EEPROMS";
char e91[]="ETAT INCONNUE";
char e92[]="ENREGISTREMENT BLOQUE";

char e911[]=" Octets Enregistrés";
char e912[]=" Octets Défectueux";

char e1011[]="VPILE=";
char e1012[]="VBAT =";

char e1111[]="MESSAGE ARGOS";
char e1112[]="PAS DES CUMULS :";

```

```

char e1113[]="{ 58 Minutes Maximum }";
char e11121[]="EN PANNE";

char e1212[]="PAS DE DONNEES.ENREGISTREMENT BLOQUE ";
char e1215[]="PAS DE PLUIE ENREGISTREE !!!";

char e1311[]="TEST DE L'AUGET";
char e1312[]="BASCULEMENTS";

char e1411[]="NUMERO SPI";
char e1412[]="HAUTEUR (mm)";
char e1413[]="TEMPERATURE (°C)";

char e1511[]="FORMATAGE EEPROMS";
char e1512[]="DECALAGE (cm):";
char e1513[]="NUMERO CAPTEUR";
char e1514[]="SEUIL D'ENREGISTREMENT (cm):";
char e1515[]="PERIODE DE MESURE DU SPI (mn):";
char e1516[]="FORMATAGE NECESSAIRE !!!";

char e1611[]="EFFACEMENT DE LA MEMOIRE";
char e1612[]="(O)ui ou (N)on";
char e1614[]="ATTENTION TOUTES LES DONNEES SERONT PERDUES...";
char e1615[]="COMMANDE IMPOSSIBLE !! TENSION PILES FAIBLES";
char e1616[]="EFFACEMENT EN COURS...";
char e1617[]="MEMOIRE EFFACEE";
char e1618[]="EFFACEMENT INCORRECT";
char e1619[]="INITIALISATION ENREGISTREMENT";
char apor[]="TAILLE DE L'AUGET";

char part1[]="PARTITION PLUVIO";
char part2[]="PARTITION LIMNI ";

char mess0[]="COMMANDES DISPONIBLES";
char mess1[]="'Q'uitter";
char mess2[]="'N'ext";
char mess3[]="'P'récedent";
char mess4[]="'S' Mesure SPI";
char mess5[]="'E'ffacement EEPROMS";
char mess6[]="'A'UGET Taille";
char mess7[]="'M'essage ARGOS";
char mess8[]="'D'ate et Heure";
char mess9[]="'H'istogramme";
char mess10[]="'T'est AUGET";
char mess11[]="'F'ormatage EEPROMS";
char mess12[]="'I'nitialisation ";

char pl[]="NOUVELLE TAILLE DE L'AUGET (Taille * 10): ";

char t25[]="25 %";
char t50[]="50 %";
char t75[]="75 %";
char t100[]="100 %";

```

PROGRAMME PRINCIPAL

TRM1 .CPP

```

/* PROGRAMME EMULATION TERMINAL TD                                     */
/* CENTRE ORSTOM MONTPELLIER                                         */
/* AUTEURS: LUGUEL OLIVIER & BAEZA YVAN                             */

/*-----HEADER-----*/

#include <graphics.h>
#include <process.h>
#include <stdlib.h>
#include <ctype.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>
#include <string.h>
#include "texte.h" /* LIBRAIRIE TEXTE */
#include "graphiqu.h" /* LIBRAIRIE GRAPHIQUE */
#include "serial.h" /* LIBRAIRIE DE GESTION RS232 */

/*-----*/

/* FONCTION PAGE_PRESENTATION                                       */
/* Affichage de la page de présentation ORSTOM                       */
/* Appel de DETECTION et FENETRE                                     */
/* Sortie visuelle écran                                           */

void Page_presentation(void)
{
detection(); /* Passage en Mode Graphique */
fenetre(10,10,630,400,BLUE);
fenetre(100,50,530,300,WHITE);
fenetre(200,150,430,200,LIGHTBLUE);

setcolor(BLACK);
setttextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(195,100,t1);

setttextstyle(TRIPLEX_FONT,HORIZ_DIR,6);
setcolor(WHITE);
outtextxy(220,140,t2);

setttextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setcolor(BLACK);
outtextxy(480,280,t3);
outtextxy(180,230,t5);
delay(4000);
}

/*-----*/

/* FONCTION ECRAN_TERMINAL                                          */
/* Dessin du terminal à l'écran                                     */
/* Sortie visuelle écran                                           */

void Ecran_terminal()
{
cleardevice();
setcolor(BLACK);
fenetre(10,10,630,40,CYAN); /* Presentation */
fenetre(10,290,630,350,BLUE); /* Message */
outtextxy(225,20,t1);
setcolor(WHITE);
}

/* PROGRAMME PRINCIPAL                                             */

```

```
void main (void)
{
Page_presentation();
Ecran_terminal();
Bouclé_programme();
}
```

il,

CHAPITRE V

GUIDE DE L'UTILISATEUR

CHAPITRE V GUIDE DE L'UTILISATEUR

I/ INTRODUCTION

TRM1 a été conçu afin de permettre le transfert et la programmation des données et des paramètres stockés en mémoire EEPROM par les centrales de type PLUVIO, LIMNI, PLUVIO-LIMNI.

Le transfert est effectué par liaison directe (RS232).

II/ MATERIEL

- Le logiciel fonctionne sur PC, PC Portable .
- Type d'écran : EGA - VGA
Format : 80 * 25
- Support : Disque dur
Disquette (250 Koctets)
- Liaison série : Liaison du type RS232
Port de communication COM1.

III/ INSTALLATION

- Introduire la disquette TRM1 dans le lecteur A de votre micro-ordinateur
- Taper la commande **TDINSTALL**
- Le chargement s'effectuera automatiquement
- Il est conseillé de faire une sauvegarde de la disquette source

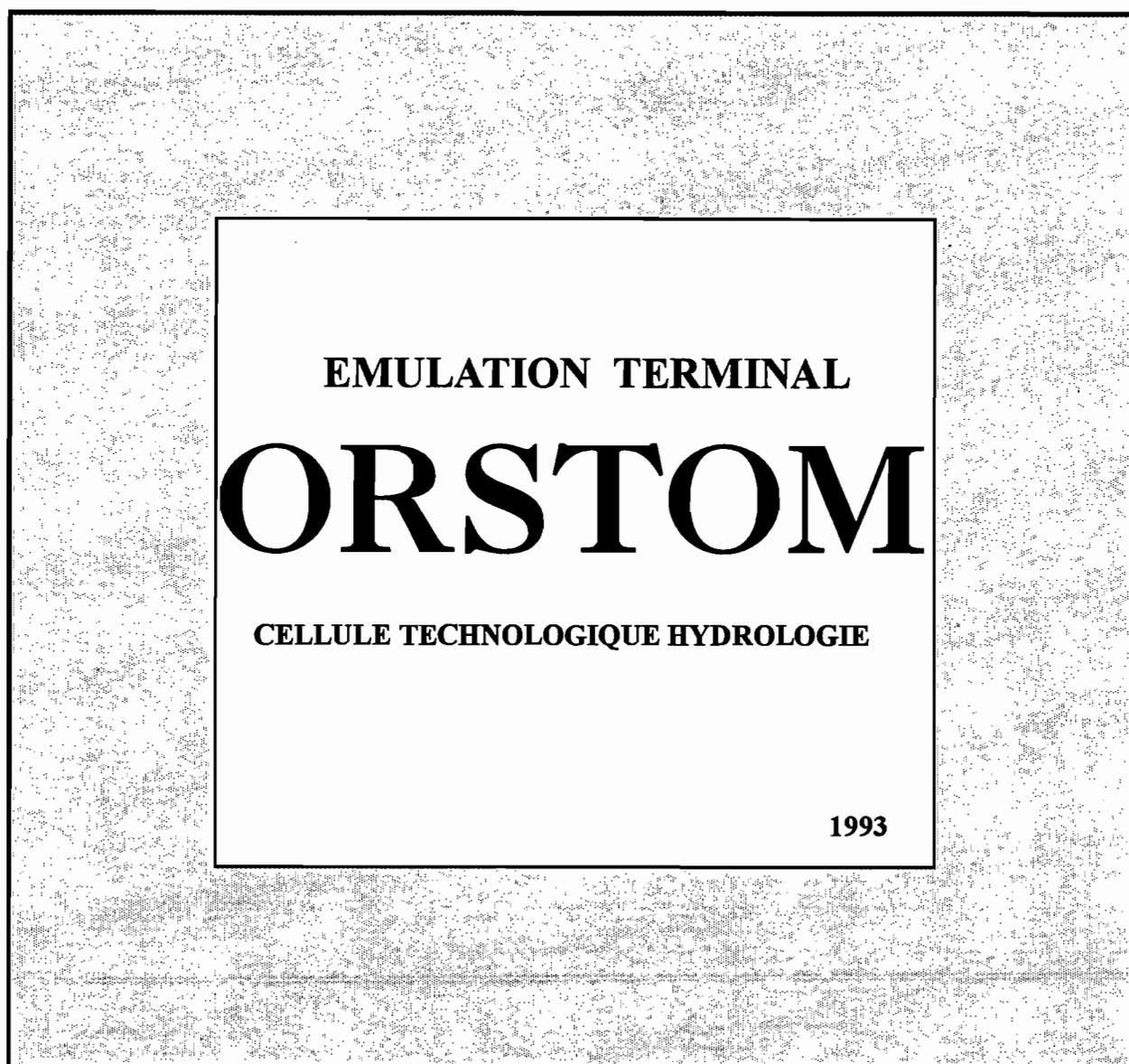
IV/ LANCEMENT DU LOGICIEL

- Placez-vous dans le répertoire de votre installation et tapez **TRM1** .

V/ DEROULEMENT DU LOGICIEL

V/1 Ecran de présentation :

Après avoir lancé le logiciel, l'écran ci-dessous apparait :

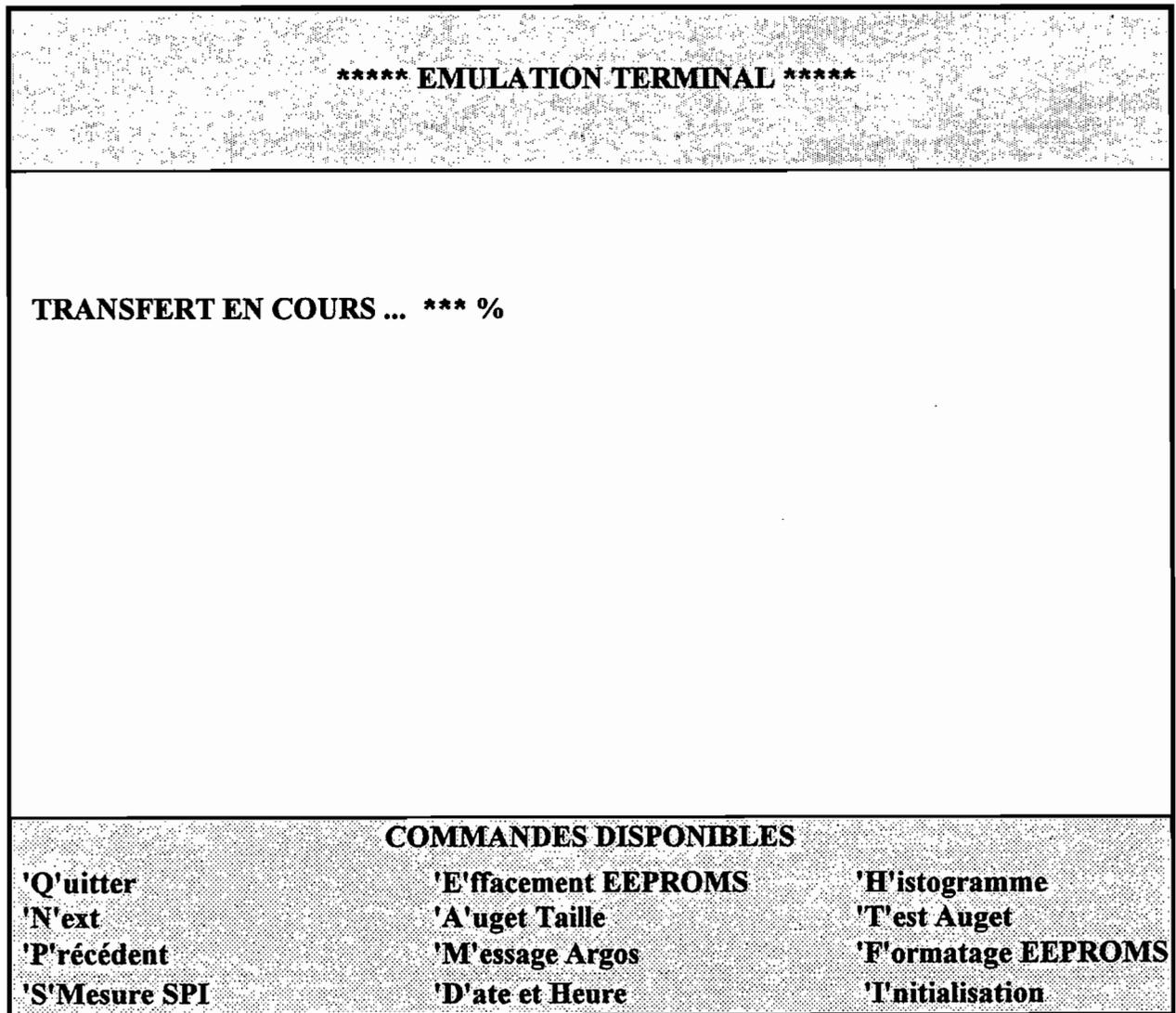


Cet écran reste environ 4 secondes .

V/2 Transfert :

Après la page de présentation, l'ordinateur envoie à la centrale plusieurs commandes afin que celle-ci lui transmette tous les paramètres de fonctionnement .

Durant cette phase, l'écran suivant est affiché :



* La durée de cette phase dépend du type de centrale, mais ne peut excéder 30 secondes .

* Si lors de l'apparition de cet écran, il n'y a pas le message (*** %), la centrale n'est pas initialisée, il faut relancer le programme .

V/3 Visualisation des Paramètres :

A cette phase du programme, la centrale a transmis tous les renseignements nécessaires à l'utilisateur et ceux-ci vont être affichés sur 2 Ecrans commutables par les touches

"P" Ecran1

"N" Ecran2

Ecran1 :

| ***** EMULATION TERMINAL ***** | | |
|--------------------------------|-------------------------------------|----------------------------|
| Type de Centrale | Etat des Tensions | Taille de l'Auget |
| Date et Heure | Dernier Basculement et Cumuls | Paramètres de Formatage |
| Durée de fonctionnement | | |
| Test Auget | | |
| COMMANDES DISPONIBLES | | |
| 'Q'uitter | 'E'ffacement EEPROMS | 'H'istogramme |
| 'N'ext | 'A'uget Taille | 'T'est Auget |
| 'P'récedent | 'M'essage Argos | 'F'ormatage EEPROMS |
| 'S'Mesure SPI | 'D'ate et Heure | 'I'nitialisation |

******* EMULATION TERMINAL *******

Mesure SPI

Message Argos

Etat des EEPROMS
Partition PLUVIO

Etat des EEPROMS
Partition LIMNI

COMMANDES DISPONIBLES

| | | |
|-----------------------|-----------------------------|----------------------------|
| 'Q'uitter | 'E'ffacement EEPROMS | 'H'istogramme |
| 'N'ext | 'A'ugment Taille | 'T'est Augment |
| 'P'récedent | 'M'essage Argos | 'F'ormatage EEPROMS |
| 'S' Mesure SPI | 'D'ate et Heure | 'I'nitialisation |

V/4 Les fenêtres :

V/4.1 Introduction

Nous venons de voir que l'on dispose de 2 Ecrans sur lesquels sont placées 12 fenêtres permettant la visualisation des paramètres de fonctionnement .

Attention les fenêtres sont fictives, elles ne sont pas encadrées mais pour la suite de notre guide nous ferons comme si c'était le cas.

Selon le type de centrale, certaines fenêtres ne seront pas présentes d'où le tableau suivant :

| Fenêtres | PLUVIO | LIMNI | PLUVIO-LIMNI |
|---------------------------|--------|-------|--------------|
| Type de Centrale | * | * | * |
| Etat des Tensions | * | * | * |
| Taille de l'Auget | * | * | * |
| Date et Heure | * | * | * |
| Durée de fonctionnement | * | * | * |
| Test Auget | * | * | * |
| Dernier Bascul et Cumul | * | | * |
| Paramètres de Formatage | | * | * |
| Mesure SPI | | * | * |
| Etat des EEPROMS (Pluvio) | * | | * |
| Etat des EEPROMS (Limni) | | * | * |
| Message ARGOS | * | * | * |

Légende : * Fenêtre présente pour cette centrale .

V/4.2 Visualisation des fenêtres :

**** Type de centrale :**

Cette fenêtre indique le numéro de série de la centrale ainsi que la version du logiciel utilisé à savoir :

{ On prend pour l'exemple une centrale du type PLUVIO }

-- Version incluant la transmission par ARGOS

SYSTEME PLUVIO
NUMERO : 18008
BALISE ARGOS INSTALLEE

-- Version n'incluant pas cette version

SYSTEME PLUVIO
NUMERO : 18008
BALISE ARGOS NON INSTALLEE

-- Version incluant la transmission par MODEM

SYSTEME PLUVIO
NUMERO : 18008
TRANSMISSION MODEM INSTALLEE

**** Etat des Tensions :**

L'utilisation d'une transmission satellite ou MODEM nécessite une alimentation par batterie . Dans ce cas 2 tensions seront affichées.

- 1- La tension des piles situées dans la centrale .
- 2- La tension de la batterie alimentant la balise satellite, ou le MODEM et la centrale .

Les tensions sont affichées sous la forme :

| | | |
|-------|---|------|
| VPILE | = | 12.1 |
| VBAT | = | 12.1 |

**** Taille de l'Auget :**

La taille de l'Auget doit être entrée lors de la mise en service du pluviomètre .

TAILLE DE L'AUGET
0.5

**** Date et Heure :**

La date et l'heure courante de la centrale sont affichées de la manière suivante :

HEURE : 09 : 13 : 57
DATE : 18 : 08 : 93

**** Durée de fonctionnement :**

La date de mise en route de la centrale ainsi que la durée de fonctionnement sont regroupées dans cette fenêtre de la manière suivante :

MISE EN ROUTE LE
18 : 08 : 93 soit
1 JOURS DE FONCTIONNEMENT

**** Test de l'Auget :**

Cette fenêtre permet de visualiser les impulsions transmises à la centrale par l'auget durant le test de ce dernier .

La visualisation à la forme :

TEST DE L'AUGET
5 BASCULEMENTS

**** Dernier Basculement et Cumul :**

La date du dernier du dernier basculement ainsi que leur cumul sont regroupés de la manière suivante :

DERNIER BASCULEMENT AU
18 : 08 à 8h56mn
CUMUL TOTAL
13 Basculements

**** Paramètres de Formatage :**

Dans cette fenêtre sont affichés tous les paramètres de fonctionnement pour les mesures du SPI, ceux-ci sont présentés de la façon suivante :

SEUIL D'ENREGISTREMENT (cm) :
1
DECALAGE (cm) :
10
PERIODE DE MESURE DU SPI (mn) :
15
NUMERO DU CAPTEUR
2002

**** Mesure du SPI :**

Cette fenêtre nous fournit le N° de la sonde interrogée, ainsi que la température et la hauteur d'eau mesurée. Toutes ces données sont affichées comme ci-dessous :

| |
|------------------|
| NUMERO SPI |
| 2002 |
| TEMPERATURE (°C) |
| 27.3 |
| HAUTEUR (cm) |
| 10.4 |

**** Etat des EEPROMS (Partition PLUVIO & LIMNI) :**

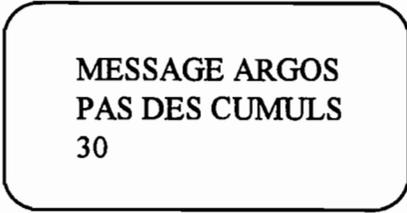
Ces deux fenêtres sont identiques, mais lorsque l'on est en présence d'une centrale du type PLUVIO-LIMNI le nombre d'octets défectueux est le même dans les deux fenêtres .

| | |
|------------------|--------------------|
| ETAT DES EEPROMS | |
| PARTITION PLUVIO | |
| 64 | Octets Enregistrés |
| 0 | Octets Défectueux |

| | |
|------------------|--------------------|
| ETAT DES EEPROMS | |
| PARTITION LIMNI | |
| 56 | Octets Enregistrés |
| 0 | Octets Défectueux |

**** Message ARGOS :**

Cette fenêtre n'apparaît que pour les centrales équipées de la transmission satellite ARGOS.
Le pas de temps des Cumuls est affiché comme suit :



MESSAGE ARGOS
PAS DES CUMULS
30

V/5 Les commandes :

V/5.1 Introduction :

Le bas des Ecrans 1&2 propose un choix de 12 commandes servant chacune à modifier ou à visualiser les paramètres de la centrale .

Attention certaines commandes ne sont pas actives sur tous les types de centrale d'où le tableau suivant :

| Commande | PLUVIO | LIMNI | PLUVIO-LIMNI |
|----------------------|--------|-------|--------------|
| 'Q'uitter | * | * | * |
| 'N'ext | * | * | * |
| 'P'récedent | * | * | * |
| 'S' Mesure SPI | | * | * |
| 'E'ffacement EEPROMS | * | * | * |
| 'A'uget Taille | * | * | * |
| 'M'essage ARGOS | * | * | * |
| 'D'ate et Heure | * | * | * |
| 'H'istogrammes | * | | * |
| 'T'est Auget | * | * | * |
| 'F'ormatage EEPROMS | | * | * |
| 'I'nitialisation | * | * | * |

V/5.2 Détails sur les commandes :

**** 'Q'uitter :**

A l'appuie sur cette touche, le processus s'interrompt et l'on retourne au DOS .

**** 'N'ext :**

Permet de passer à l'Ecran2 .

**** 'P'récedent :**

Permet de passer à l'Ecran1 .

**** 'S' Mesure SPI :**

Quelque soit l'Ecran dans lequel on se trouve, l'appuie sur cette touche nous renvoie directement à l'Ecran2 .

La fenêtre concernant la Mesure SPI est alors la suivante :

```
NUMERO SPI
2002
TEMPERATURE (°C)
??????
HAUTEUR (cm)
??????
```

Après environ 4 secondes, la nouvelle température et la nouvelle hauteur sont affichées .

**** 'E'ffacement des EEPROMS :**

Cette commande sert à remettre à zéro toutes les données enregistrées dans les mémoires ainsi qu'à réinitialiser la centrale .

Ce processus se décompose en plusieurs phases :

a) Confirmation de la demande d'Effacement

EFFACEMENT DE LA MEMOIRE
(O)ui ou (N)on

Si l'on répond 'N', alors on revient à l'Ecran1, sinon on passe à la phase b .

b) Effacement

EFFACEMENT DE LA MEMOIRE
(O)ui ou (N)on

EFFACEMENT EN COUR

La durée de cette phase varie selon le contenu des EEPROMS, mais ne peut excéder 3 mn .

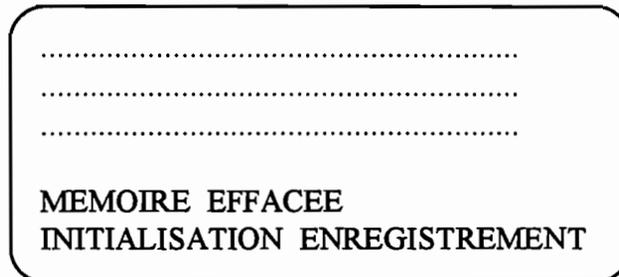
A ce niveau 2 cas peuvent se produire :

c) Le processus s'est mal déroulé, alors

EFFACEMENT INCORRECT

Retour à l'Ecran1 .

d) Le processus s'est bien déroulé



Retour Ecran1 .

**** 'A'uget Taille :**

Cette commande permet de modifier la taille de l'auget qui peut varier de 0.1 à 1 .
La valeur entrée sera égale à la taille réelle * 10 .

La demande se fera sous la forme suivante :

NOUVELLE TAILLE DE L'AUGET (TAILLE * 10) :

Retour à l'Ecran1 .

**** 'M'essage ARGOS :**

Cette commande sert à modifier la valeur du pas de temps pour les cumuls destinés au satellite ARGOS .

MESSAGE ARGOS
PAS DES CUMULS :
{ 58 Minutes maximum }

Si l'on entre une mauvaise valeur, alors le message suivant apparait :

MAUVAISE TAILLE !!!

Sinon l'on revient directement à l'Ecran2 .

**** Date et Heure :**

Cette commande permet de modifier la date et l'heure de la centrale .
Le processus se décompose en plusieurs phases, mais l'écran terminal est le suivant :

| | | |
|----------|---|-------|
| Heure | : | |
| Minutes | : | |
| Secondes | : | |
| Jour | : | |
| Mois | : | |
| Année | : | |

Si l'on entre une valeur erronée, alors le message suivant apparait :

DATE INCORRECTE

Sinon 2 cas peuvent se produire :

- a) La centrale n'est pas bien programmée

HORLOGE EN PANNE

Retour à l'Ecran1 .

b) La programmation s'est bien déroulée

SYNCHRO OK

Retour à l'Ecran1 .

**** 'H'istogramme :**

Cette commande sert à la visualisation du cumul de pluie tombée à une date précise.
Ce processus se décompose en plusieurs phases :

a) Vérification :

Dans cette phase on vérifie que la centrale n'est pas en mode Enregistrement Bloqué,
sinon

PAS DE DONNEES . ENREGISTREMENT BLOQUE

Retour à l'Ecran1 .

b) Saisie de la date recherchée :

Jour : Mois : Année :

Si la date est incorrecte, alors le message suivant apparait

DATE INCORRECTE

c) Recherche :

Si il n'y a pas d'enregistrements, le message est le suivant

PAS DE PLUIE ENREGISTREE !!!

Sinon :

TRANSFERT EN COURS

Ce message reste environ 2 à 3 secondes, après nous obtenons la date trouvée qui est la plus proche de la date demandée ainsi que le cumul pour ce jour .

DATE : 18 : 08 : 93
CUMUL TOTAL 150 (1/10 mm)

A ce niveau plusieurs possibilités s'offre à nous :

'Q'uitter

'P' Jour précédent

'S' Jour suivant

'H' Détail heure par heure

*** 'Q'uitter :**

Cette micro-commande sert à sortir du mode histogramme et à revenir à l'Ecran1 .

*** 'P' Jour précédent :**

La centrale nous transmet la date et le cumul du jour précédent celui en cours, l'affichage et le même que ci-dessus .

Pendant la phase de Recherche, le message suivant est affiché :

TRANSFERT EN COURS

Si la centrale n'a rien trouvé, alors l'affichage sera le suivant :

DATE : 00 : 00 : 00
CUMUL TOTAL 0 (1/10 mm)

*** 'S' Jour suivant :**

Le processus est identique à celui de la micro-commande 'P', mais pour le jour suivant .

*** 'H' Détail heure par heure :**

Cette micro-commande sert à visualiser, pour le jour en cours, les cumuls heure par heure .
L'affichage aura la forme suivante :

| | | | | | | |
|-------------|---|-----|---|-----------|---|----|
| DATE | : | 18 | : | 08 | : | 93 |
| CUMUL TOTAL | | 150 | | (1/10 mm) | | |
| 8h à 9h | | 65 | | (1/10 mm) | | |
| 9h à 10h | | 20 | | (1/10 mm) | | |
| 10h à 11h | | 65 | | (1/10 mm) | | |

**** 'T'est Auget :**

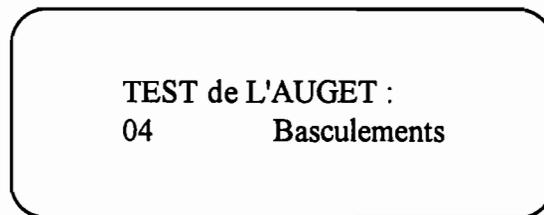
Lorsque cette commande est frappée, le programme transmet une commande à la centrale lui indiquant de se mettre en mode de Test .

Dans ce mode, chaque basculement détecté sera transmis au PC et ne sera pas enregistré dans la cartouche d'EEPROMS .

Le nombre de basculements est affiché sur l'écran et s'incrémente à chaque transmission de l'enregistrement .

Cette commande permet ainsi de tester le contact du pluviographe et de calibrer l'auge .

L'écran est le suivant :



TEST de L'AUGET :
04 Basculements

Dans ce mode, l'horloge est arrêtée car l'ordinateur scrute sans arrêt le clavier et le port série.

Pour sortir de ce mode, il suffit d'appuyer sur une touche quelconque, la centrale est alors remise en mode Enregistrement .

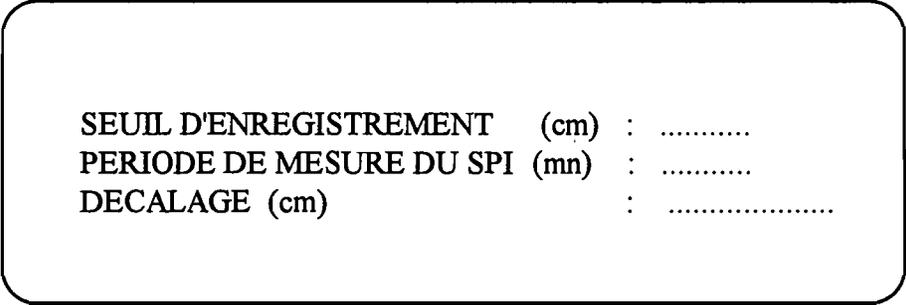
**** 'F'ormatage EEPROMS :**

Cette commande permet de modifier le seuil d'enregistrement, la période de mesure et le décalage pour les mesures du SPI .

Les valeurs limites sont :

| | | |
|----------|-------|----|
| Seuil | < 99 | cm |
| Période | < 99 | mn |
| Décalage | < 999 | mm |

L'écran est le suivant :



```
SEUIL D'ENREGISTREMENT (cm) : .....  
PERIODE DE MESURE DU SPI (mn) : .....  
DECALAGE (cm) : .....
```

Retour à l'Ecran1 .

**** 'Initialisation :**

Cette commande permet de redemander à la centrale tous les paramètres de fonctionnement comme pour la phase de démarrage .

L'écran est alors le même que durant le lancement du programme .

****** Remarque :**

Dans l'utilisation de toutes les commandes, si la tension des piles est trop faible, celles-ci ne peuvent être exécutées et le message suivant apparait :

COMMANDE IMPOSSIBLE !!! TENSION PILES FAIBLES

CONCLUSION

CONCLUSION

Le programme qui vient d'être réalisé fonctionne pour les 3 types de centrales définies précédemment.
Il a été écrit en Turbo C et nous a donc apporté une connaissance approfondie de ce langage .

Pour arriver au programme final, nous avons donc dû faire une analyse importante qui nous a ralenti dans la progression du programme .

Il reste un point à traiter pour que ce dernier soit optimisé, il faudrait créer un fichier historique dans lequel seraient stockés tous les paramètres de fonctionnement de la centrale avant et après une quelconque modification .

Cette partie fera l'objet d'un prochain stage .

Pour notre part, le programme développé est autonome mais il peut également s'intégrer dans un logiciel déjà existant tel que VISUAL qui sert à tracer des graphes en fonction des données stockées dans les centrales .

ANNEXES

93, Route de Corbeil
91700 STE-GENEVIEVE-DES-BOIS
Tél. : (1) 69 04 93 93
Télex : 250304 F

NOTE TECHNIQUE ENREGISTREUR PLUVIO 89



PLUVIO 89 est un enregistreur autonome destiné à être connecté à un pluviomètre à augets basculeurs équipé d'un contacteur.

Le support de mémorisation est constitué par 4 mémoires fixes effaçables électriquement (Technologie EEPROM). Ce support statique et non-volatile est particulièrement bien adapté aux environnements sévères.

PLUVIO 89 enregistre systématiquement la date de chaque impulsion générée par l'auget du pluviomètre. Cet histogramme permet, lors du dépouillement de la mémoire, de reconstituer un pluviogramme et de traiter les intensités de pluie sur des périodes de temps variables.

PLUVIO 89 possède une base de temps calendaire. La date de chaque basculement est exprimée en heures, minutes, secondes, le changement de jour en année, mois, jour.

Une capacité mémoire de 32 kilo-octets permet le stockage de 11 000 impulsions soit, pour un auget de 0,5 mm, un équivalent "hauteur d'eau" de 5,50 m.

De façon à renseigner l'opérateur sur les différents paramètres de fonctionnement et des données enregistrées, un terminal de poche muni d'un clavier et

- * *SYSTEME AUTONOME D'ENREGISTREMENT ET DE TRANSMISSION D'INFORMATIONS PLUVIOMETRIQUES*
- * *SE CONNECTE A TOUT PLUVIOMETRE A AUGET DELIVRANT DES IMPULSIONS.*
- * *ENREGISTREMENT SUR MEMOIRE FIXE ET NON VOLATILE (technologie EEPROM)*
- * *CAPACITE DE STOCKAGE IMPORTANTE : 11 000 IMPULSIONS DATEES*
- * *BASE DE TEMPS CALENDRAIRE*

EN OPTION :

- * *TERMINAL DE POCHE TD 89*
- * *INTERFACE POUR EMETTEUR SATELLITE ARGOS*
- * *LIAISON MODEM (en développement)*

d'un affichage alpha numérique se connecte à l'enregistreur. Il permet d'accéder aux fonctions suivantes :

1) AFFICHAGE :

- *N° de série, état du système.*
- *Durée de fonctionnement, date du dernier basculement et Cumul.*
- *Histogramme en mm de pluie par jour, possibilité de "Zoom" sur les heures.*
- *Etat de remplissage de la mémoire.*
- *Visualisation des tensions (piles, batterie).*

2) MISE A L'HEURE DE L'HORLOGE

3) POSSIBILITE D'EFFACER LA MEMOIRE ET DE REINITIALISER L'ENREGISTREUR

PLUVIO 89 est muni d'une liaison RS232C lui permettant d'être connecté à n'importe quel micro-ordinateur de façon à dépouiller les données enregistrées et d'accéder à toutes les fonctions du Terminal.

PLUVIO 89 a été conçu pour s'interfacer à un émetteur satellite ARGOS. En option, il peut être intégré dans une armoire étanche et verrouillable renfermant la batterie, l'émetteur, et supportant l'antenne ARGOS.

CARACTERISTIQUES :

*** ENTREE CAPTEUR**

- 1 voie acceptant un signal impulsionnel actif au Ov, filtré et mémorisé par logiciel.

*** SORTIE**

- 1 voie RS232C pour la connexion :
 - . du terminal d'initialisation TD89 ou,
 - . d'un micro-ordinateur ou,
 - . d'un émetteur satellite ou,
 - . d'un modem.

*** ALIMENTATION**

- De 8 v à 16 v (2 mA en veille, 100 mA en écriture)
- Piles internes (Durée 4 mois)
- Possibilité de connecter une batterie externe.

*** ENVIRONNEMENT CLIMATIQUE**

- Température : -20° à +70°C
- Humidité : 100 %
- Coffret étanche selon la norme IP65

*** BASE DE TEMPS**

- Horloge temps réel initialisable
- Dérive inférieure à 1 mn par mois

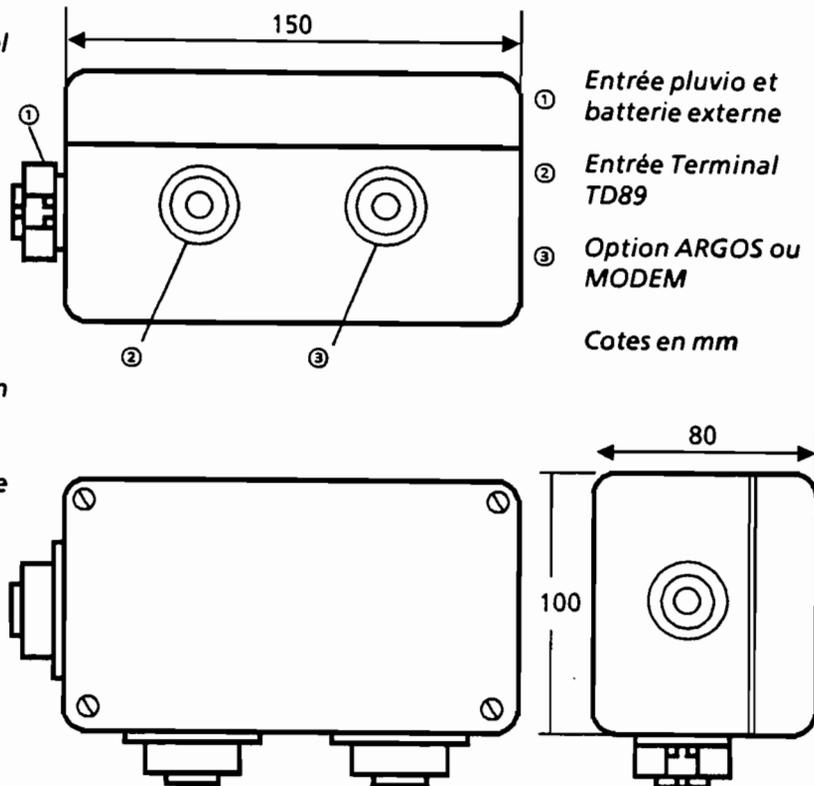
*** CONNEXIONS**

- Prises étanches type AMPHENOL C16

*** ENREGISTREMENT**

- Sur mémoires EEPROM internes (32 Ko)
- Capacité de stockage : 11000 basculements

PLAN MECANIQUE



*** OPTIONS**

- Interface pour émetteur ARGOS ou METEOSAT
- Armoire de conditionnement polyester
- Panneau solaire 8Wh avec régulateur
- Batterie sans entretien

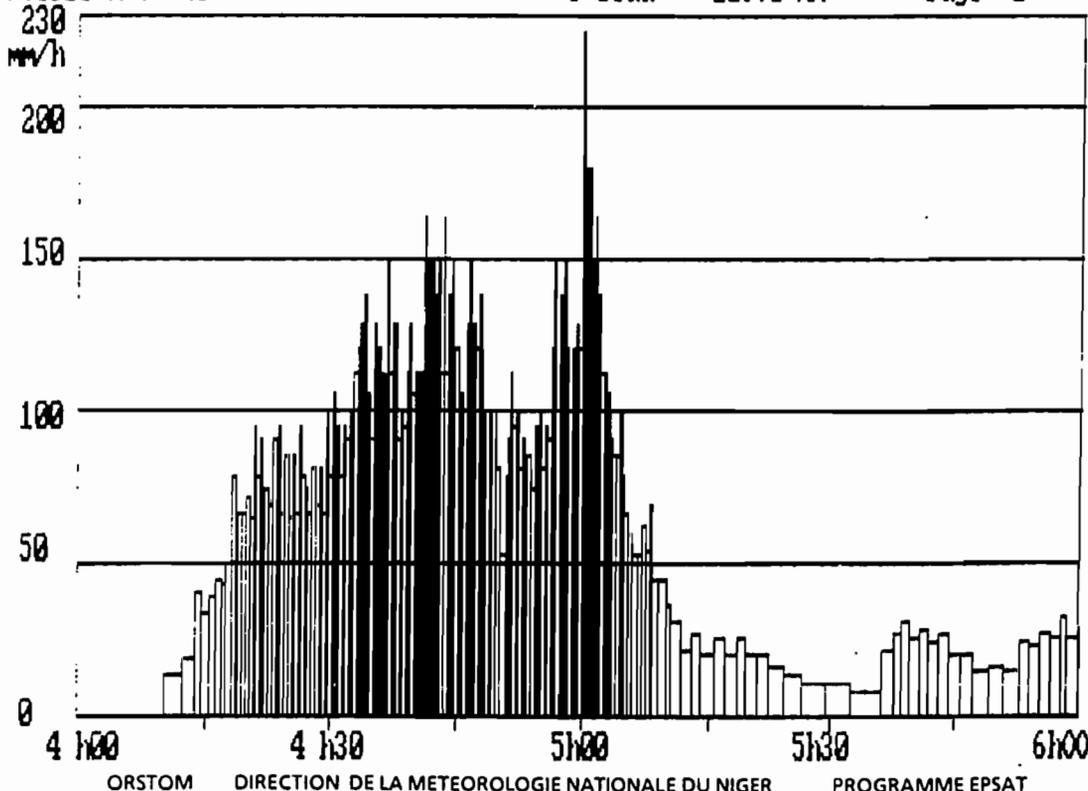
STATION : NIAMEY POUDDRIERE

Averse N°: 16

DEBUT D' AVERSE LE 04/08/1989 A 04h00'55

P seau = 119.0 mm Page 1

Type de sortie pouvant être obtenue à partir d'un enregistreur PLUVIO 89



93, Route de Corbeil
91700 STE-GENEVIEVE-DES-BOIS
Tél. : (1) 69 04 93 93
Télex : 250304 F



Développée en collaboration avec le service hydrologique de l'ORSTOM(*), le SPI III est un ensemble électronique et informatique destiné à la mesure précise des niveaux et températures des liquides de densité constante.

La hauteur d'eau et la température sont mesurées par un capteur de pression délivrant un signal électrique fonction de la pression hydrostatique du liquide par rapport à la pression atmosphérique. Ce capteur est constitué d'une jauge piézo-résistive conditionnée dans un boîtier inox et isolée du liquide par un diaphragme en alliage platine tantale.

La sonde SPI III contient un micro-processeur gérant son fonctionnement et réalisant les acquisitions capteurs, les traitements de mise à l'échelle et de compensation thermiques de façon à élaborer des résultats en vraie grandeur incluant la hauteur d'eau et la température.

Ces informations constituent un message binaire qui est transmis par une liaison série basse vitesse, ou RS232C, isolée par des coupleurs opto-électroniques et protégée contre la foudre.

Ce concept original permet une transmission sur plusieurs kilomètres sans dégradation de la précision ou de la sensibilité de la mesure.

(*) Institut Français de Recherche Scientifique pour le Développement en Coopération

NOTE TECHNIQUE

SONDE SPI III

- * SONDE LIMNIMETRIQUE IMMERGEE
- * CABLE DEMONTABLE
- * CAPTEUR DE PRESSION PIEZO-RESISTIF
- * MESURE DE LA HAUTEUR D'EAU ET DE LA T°
- * TRANSMISSION BINAIRE EN VRAIE GRANDEUR
- * LIAISON SERIE SYNCHRONNE BASSE VITESSE PERMETTANT UNE TRANSMISSION SUR PLUSIEURS KILOMETRES
- * LIAISON ASYNCHRONNE TYPE RS232C
- * ZERO ET FACTEUR D'ECHELLE TELECHARGEABLES
- * FAIBLE CONSOMMATION, DIMENSION REDUITE

EN OPTION :

- * CABLE DE LIAISON SUPPLEMENTAIRE
- * ACCESSOIRES DE FIXATION

CONCEPTEURS :

ORSTOM (MM. COLOMBANI & FRITSCH)
ELSYDE (MM. MAILLACH & SEVEQUE)

Les sondes sont toutes interchangeables et toutes compatibles sans aucune adaptation avec les centrales d'enregistrement CHLOE.

Chaque sonde SPI III est calibrée en pression et température et ses courbes d'étalonnages sont stockées dans la mémoire de son micro-processeur. Le zéro et le facteur d'échelle peuvent être télé-chargés par la liaison.

Le SPI III est activé sur demande du système d'acquisition. Le temps de mesure est environ 3 secondes et la transmission dure 640 ms.

Le câble de liaison conçu spécialement contient un capillaire de mise à la pression atmosphérique et les conducteurs électriques. Ce câble, blindé par une tresse métallique et gainé d'une épaisseur de 2 mm de polyuréthane se termine par un connecteur Jupiter coté sonde, coté centrale il se raccorde à un échangeur déshydratant évitant la condensation interne du capillaire (connecteur étanche Amphénol C16).

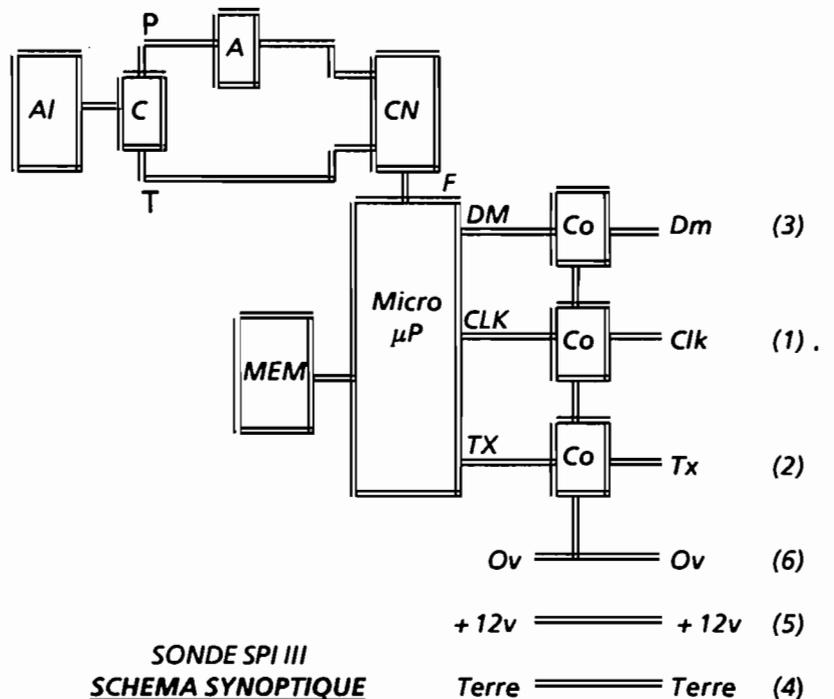
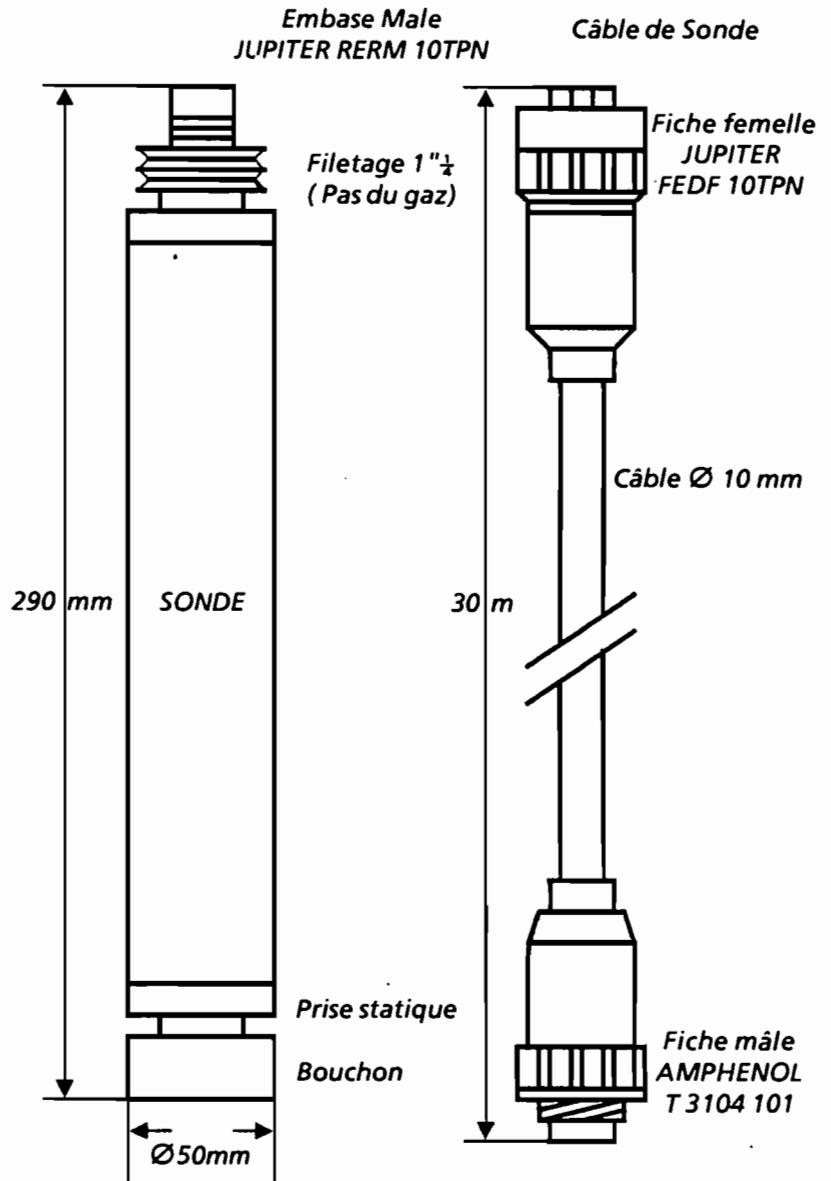
La sonde SPI III est calibrée en standard pour une étendue de mesure de 10 mètres. Elle peut être fixée en dessous des plus basses eaux ou suspendue dans un puits limnimétrique conventionnel. La géométrie de sa prise statique lui assure une bonne qualité de mesure dans des écoulements rapides ou turbulents.

CARACTERISTIQUES

- * **ETENDUE DE MESURE** : 0 à 10 m
- * **SURCHARGE OCCASIONNELLE** : 15 m
- * **PRECISION DES MESURES** : (sous 12,5 v)
 - Niveau : ± 10 mm
 - Température : ± 1°C
- * **TEMPERATURE D'UTILISATION** :
 - Immergée : - 20° à + 70°C
 - Immersée : 0° à + 50°C
- * **VITESSE DE POURSUITE** : 1 cm/sec MAX
- * **TEMPS DE MESURE** : environ 3 s
- * **COURANT CONSOMME** : 5 mA moyen (sous 12 v)
- * **SORTIE** :
 - Transmission série synchrone 100 bits/sec en boucle de courant
 - Transmission type RS232C
- * **LONGUEUR MAXIMUM DE LA LIAISON** : Correspondant à une constante de temps d'un circuit équivalent RC : 3 ms
- * **DIMENSIONS** : Ø 50 mm : L 300 mm
- * **MATIERE** : PVC
- * **CONNECTIQUE** : Embase JUPITER PERM 10 TPN
- * **POIDS** : 1 kg (Sonde seule)
- * **CABLE DE LIAISON DEMONTABLE** : L 30 m
- * **ACCESSOIRES** :
 - Rallonges de 10 à 50 m
 - Lest en plomb s'adaptant sur la tête inférieure.

- Al = Alimentation capteur
- C = Capteur piézo-résistif
- P = Signal pression
- T = Signal température
- A = Amplificateur
- CN = Conversion numérique
- F = Signal capteur en fréquence
- DM = Demande de mesure
- CLK = Horloge de transmission
- TX = Données
- MEM = Mémoire logiciel + Courbes d'étalonnage
- µP = Microprocesseur
- Co = Coupleur opto-électronique

PLAN MECANIQUE



SONDE SPI III
SCHEMA SYNOPTIQUE

93, Route de Corbeil
91700 STE-GENEVIEVE-DES-BOIS
Tél. : (1) 69 04 93 93
Télex : 250304 F



Le Terminal Multifonctions TD87 est un outil d'aide au diagnostic pour le personnel d'exploitation des centrales limnimétriques.

Le TD87 se connecte entre la centrale CHLOE et la sonde SPI. Il est pourvu des fonctions suivantes :

- Fonction **ESPION** :

ESPION 01 OK
3210 H2510 T09.5

Visualisation des données transmises par la sonde SPI à la centrale lors d'une mesure.

- Fonction **TRANSMISSION** :

TRANS 01
3210 H2510 T09.5

Le Terminal joue le rôle de la sonde SPI. Il transmet à la centrale des données choisies par l'utilisateur de façon à vérifier le bon fonctionnement de celle-ci.

- Fonction **MESURE** :

MESURE 01 OK
3210 H2510 T09.5

La sonde SPI est interrogée en continu à raison d'une mesure toutes les 5 secondes. Le TD87 jouant le rôle de la centrale.

- Fonction **VOLTMETRE** :

TENSION : 12.3 V

Il affiche la valeur de la tension de ligne.

NOTE TECHNIQUE

TERMINAL TD 87

- * AFFICHAGE LCD DE 2 LIGNES DE 16 CARACTERES ALPHA-NUMERIQUES. CLAVIER 12 TOUCHES
- * TRANSMISSION SERIE SYNCHRONE 500 BIT/SEC ISOLEE PAR OPTO COUPLEURS
- * VISUALISATION DE LA TRANSMISSION PAR DIODES LED
- * COFFRET PLASTIQUE AVEC CAPOT DE PROTECTION

- Fonction **REGLAGE CAPTEUR** :

REGLAGE CAPTEUR
CODE : 00000

Permet de régler le zéro et le facteur d'échelle de la sonde. L'accès aux réglages est protégé par un code à 5 chiffres.

L'accès étant autorisé :

REGLAGE DU ZERO
OUI = 0 NON = VAL

Le réglage du point zéro du capteur doit être effectué la sonde hors d'eau.

REGLAGE DU COEFF
C = 0.0000

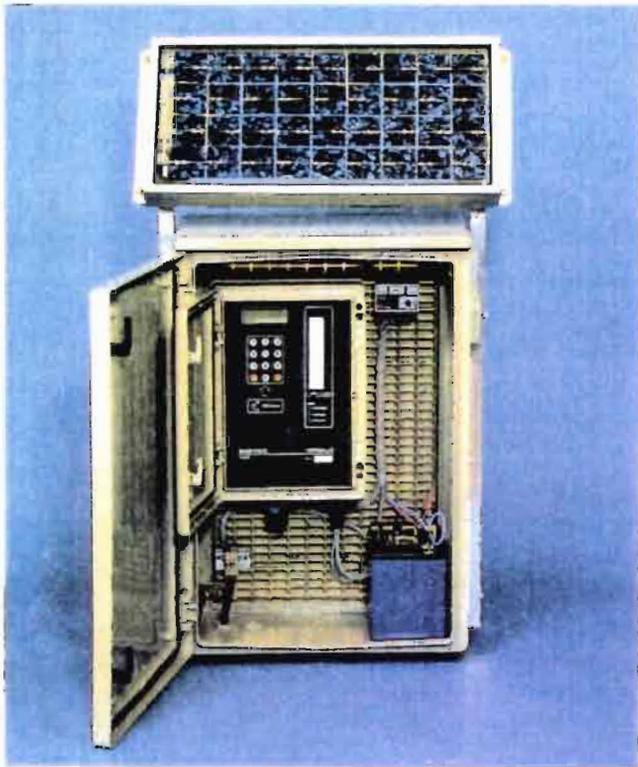
0.8 < COEFF. < 1.2

CARACTERISTIQUES :

- * Tension d'alimentation : 8 à 15 v
- * Courant consommé : 10 mA
- * Température d'utilisation : - 10° à + 50°C
- * Humidité relative : 90 % Max.
- * Dimensions : 200 x 130 x 50 mm
- * Poids : 500 g

93, Route de Corbeil
91700 STE-GENEVIEVE-DES-BOIS
Tél. : (1) 69 04 93 93
Télex : 250304 F

NOTE TECHNIQUE CENTRALE CHLOE-D



CHLOE-D est une centrale autonome réalisant l'acquisition et l'enregistrement de la hauteur et de la température de l'eau mesurées par la sonde limnimétrique SPI III. Cette centrale est aussi équipée d'une entrée pluviométrique.

Le support de mémorisation est constitué par une cartouche amovible de grande capacité effaçable électriquement et réutilisable (technologie EEPROM).

Ce support statique et non-volatile est particulièrement bien adapté aux environnements sévères et à l'utilisation sur le terrain sans précautions particulières.

Les mesures sont réalisées à pas de temps réglable (de 01 à 99 mn) et la hauteur d'eau est mémorisée si elle représente une variation significative réglable de 01 à 99 cm. CHLOE-D réalise une saisie intelligente des données proportionnelle à la variation du phénomène à enregistrer.

CHLOE-D est équipée d'une base de temps calendrier précise. Le clavier et l'afficheur à cristaux liquides situés en face avant permettent d'accéder aux fonctions suivantes :

- * SYSTEME AUTONOME D'ENREGISTREMENT ET DE TRANSMISSION D'INFORMATIONS LIMNIMÉTRIQUES, PLUVIOMÉTRIQUES ET DE TEMPÉRATURE
- * COMPATIBLE AVEC LA SONDÉ LIMNIMÉTRIQUE SPI III
- * ENREGISTREMENT SUR CARTOUCHE AMOVIBLE CEE 64 (technologie EEPROM)
- * TEMPS D'ECHANTILLONNAGE DE 01 à 99 mn
- * ENREGISTREMENT PILOTE PAR LA VARIATION DE LA HAUTEUR D'EAU (01 à 99 cm)
- * BASE DE TEMPS CALENDRAIRE
- * CLAVIER D'INITIALISATION ET AFFICHAGE ALPHANUMÉRIQUE

OPTIONS :

- * CARTE ENTRES SUPPLEMENTAIRES 0-5v
- * INTERFACE POUR EMETTEUR SATELITE ARGOS

- Initialisation de l'horloge
- Décalage de la hauteur d'eau
- Visualisation des mesures en vraie grandeur
- Formatage de la cartouche
- Mesures des tensions batterie et panneau
- Initialisation du Seuil d'enregistrement et de la Période.

L'autonomie de la cartouche mémoire dépend du temps d'échantillonnage et du seuil de sensibilité choisis.

Une saisie systématique toutes les 10 minutes assure une autonomie d'environ 4 mois.

Les cartouches CEE64 sont dépouillées sur tout ordinateur possédant une sortie RS232C. Le transfert est effectué par une interface spéciale (LCM) dans laquelle s'insère la cartouche. Après transfert la cartouche peut être effacée sur ce même appareil.

CHLOE-D a été conçu pour s'interfacer à un émetteur satellite ARGOS.

En option, CHLOE-D peut être intégrée dans une armoire étanche et verrouillable renfermant la batterie et supportant l'antenne ARGOS.

CARACTERISTIQUES :*** ENTRES CAPTEURS**

- 1 voie limnimétrique pour sonde SPI III
- 1 voie acceptant un signal impulsionnel actif au Ov, filtré et mémorisé par logiciel.

*** SORTIE**

- Enregistrement sur cartouche amovible CEE64
- 1 vecteur de transmission série compatible avec émetteurs ARGOS (UHF88) ou METEOSAT

*** ALIMENTATION**

- De 10.5 v à 16 v (6 mA en veille, 300 mA en écriture)

*** ENVIRONNEMENT CLIMATIQUE**

- Température : -20° à +70°C
- Humidité : 95 %
- Coffret étanche selon la norme IP55

*** BASE DE TEMPS**

- Horloge temps réel initialisable
- Dérive inférieure à 1 mn par mois

*** CONNEXIONS**

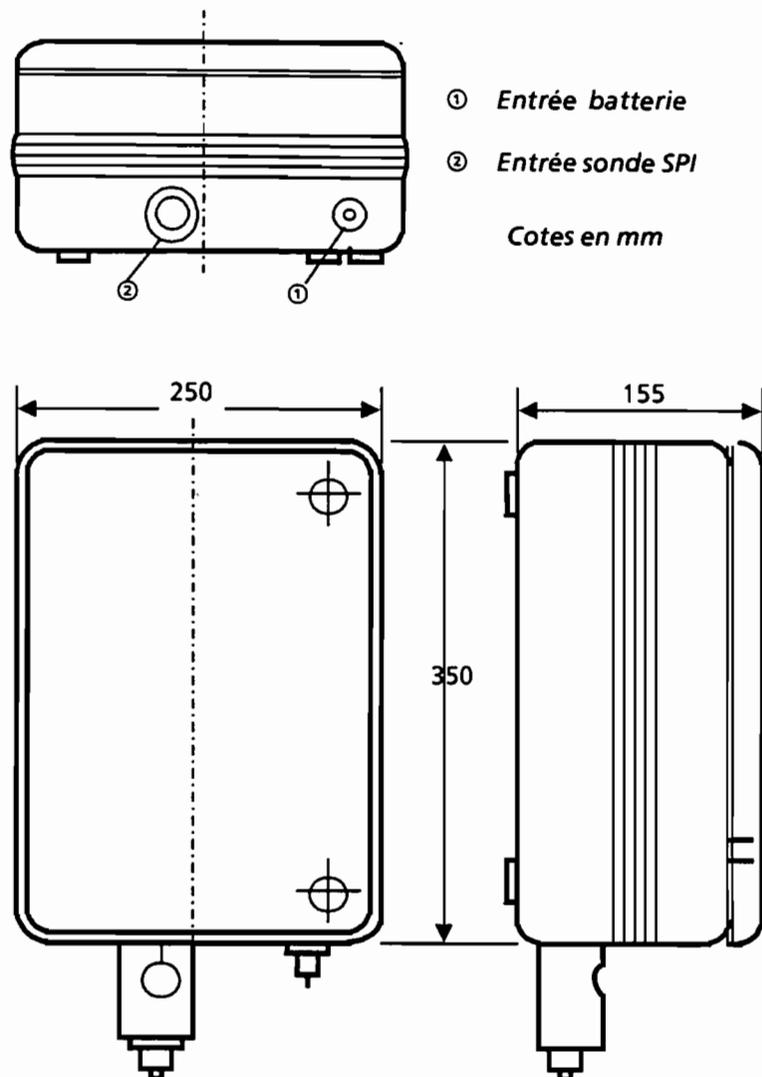
- Prises étanches type AMPHENOL C16

*** ENREGISTREMENT**

- Sur cartouche EEPROM amovible CEE64
- Rythme d'acquisition réglable (de 01 à 99 mn)
- Seuil significatif réglable (de 01 à 99 cm)

*** OPTIONS**

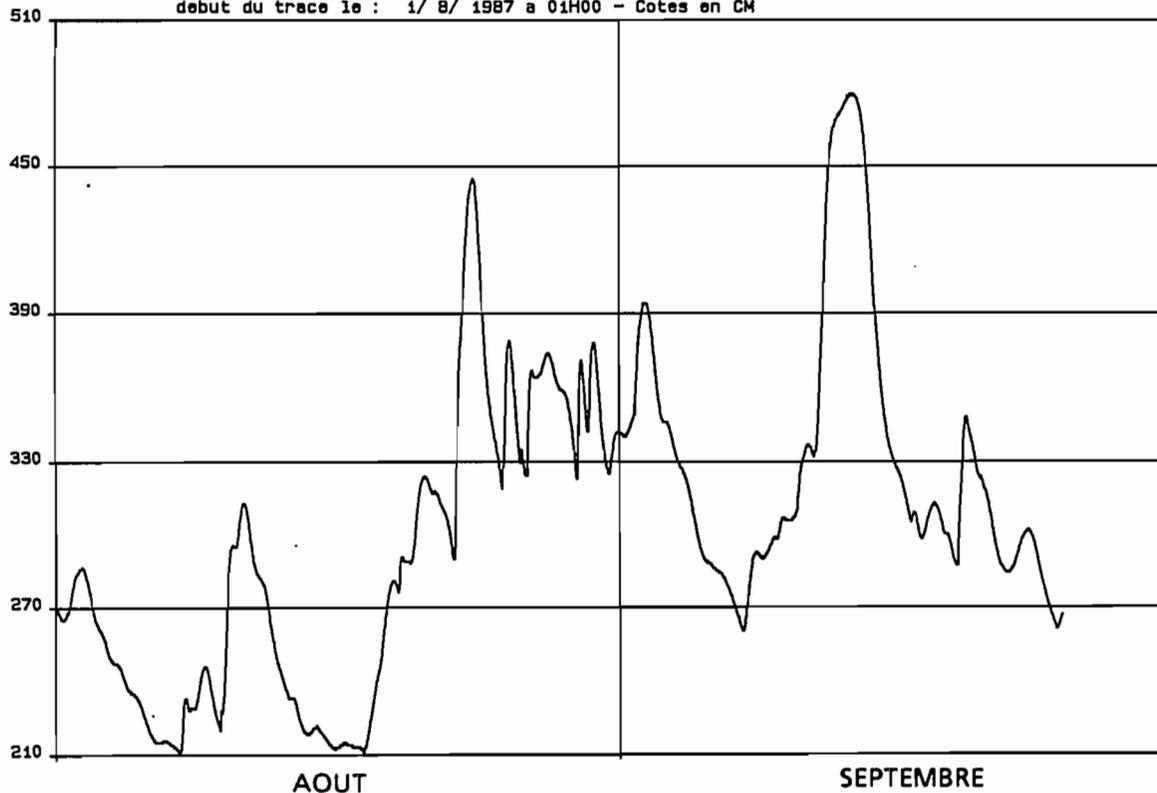
- Interface pour émetteur ARGOS ou METEOSAT
- Armoire de conditionnement polyester
- Panneau solaire 8Wh avec régulateur
- Batterie sans entretien

PLAN MECANIQUE

1171503509-1 DION a DIAMARADOU

debut du trace le : 1/ 8/ 1987 a 01H00 - Cotes en CM

Dépouillement
d'une cartouche
réalisée par
le logiciel
"HYDROM"

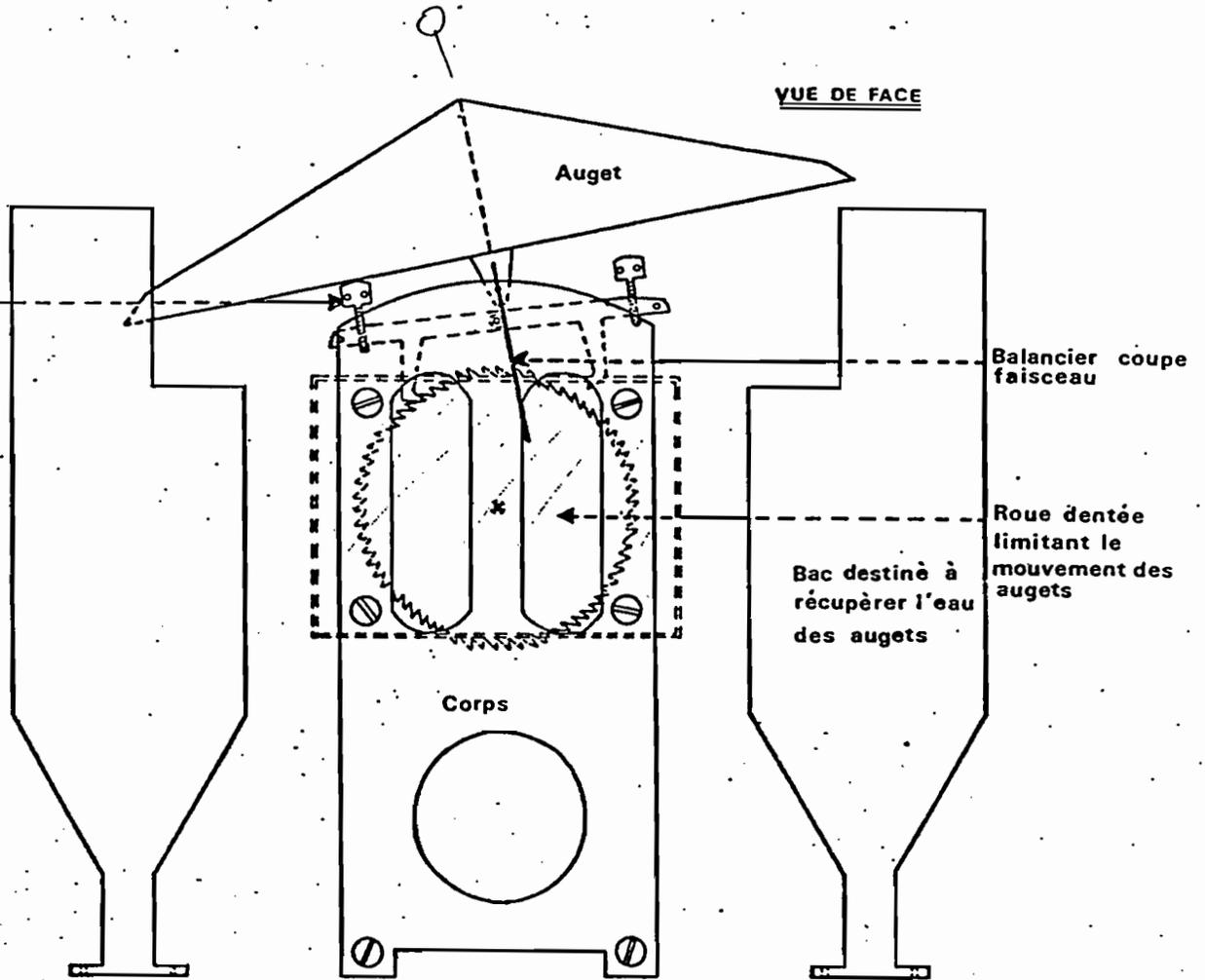


Pluviographe à augets basculeurs "Précis Mécanique"

Positionnement de la plaque de circuit imprimé (double trait pointillé)
et du balancier coupe faisceau (trait fort)

VUE DE FACE

Vis de réglage du
volume de remplissage
des augets



Echelle : $1/\sqrt{2}$

Roue dentée

Auget

Vis de réglage

Balancier

Photo coupleur

Plaque de circuit imprimé

VUE DE PROFIL

Corps

J.-M. LAPETITE

Schéma électronique d'un détecteur de basculements des augets d'un pluviographe "Précis Mécanique"

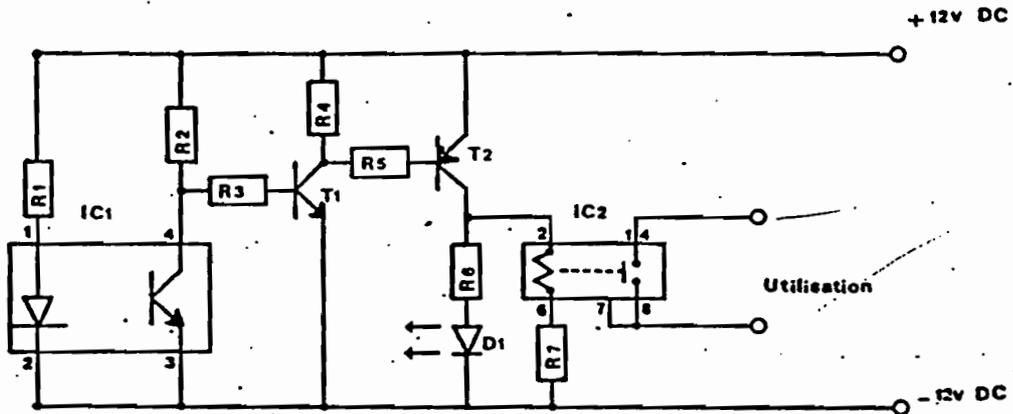
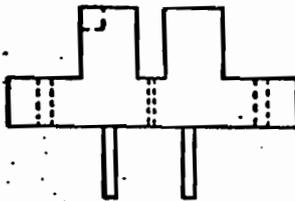
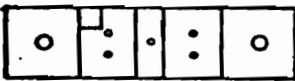


Photo coupleur à fenêtre

Echelle: 1x/2



Nomenclature

Résistances

- R1 : 1 Kohm
- R2 : 150 Kohms
- R3 : 150 Kohms
- R4 : 10 Kohms
- R5 : 10 Kohms
- R6 : 1 Kohm
- R7 : 150 ohms

Transistors

- T1 : 2N2369
- T2 : 2N924

Circuits Intégrés

- IC1: Photo Coupleur MCT8
- IC2: Celduc D31C2150

Divers

- D1 : Led Rouge Ø 3 mm