

PROCOLE
DE COMMUNICATION

AFFICHEURS
DE LA GAMME LEDI

(sauf LEDI GRAPH et LEDI SIGN)

SOMMAIRE

1. GENERALITES	1
1.1. LES TYPES DE DONNEES STOCKES DANS L’AFFICHEUR	1
1.1.1. <i>Les données système</i>	1
1.1.2. <i>Les données utilisateurs</i>	1
1.1.3. <i>Organisation de la mémoire</i>	1
2. LE PROTOCOLE DE COMMUNICATION.....	2
2.1. IDENTIFICATION DE L’AFFICHEUR:.....	2
2.2. CONSTITUTION DU MESSAGE	2
2.3. LE TIMING.....	3
2.3.1. <i>Le système RS485</i>	3
3. CREATION D’UN PROGRAMME.....	4
3.1. GENERALITE	4
3.2. EMISSION EN PLUSIEURS PAQUETS	4
3.3. LES PRINCIPAUX ORDRES.....	4
3.4. HORODATAGE D’UN MESSAGE	6
4. EXEMPLES DE TRANSMISSIONS :	7
4.1. CREATION D’UN MESSAGE.....	7
4.2. SUPPRESSION D’UN MESSAGE	8
4.3. INITIALISATION D’UN AFFICHEUR.....	8
4.4. ARRET DE L’AFFICHEUR	8
4.5. MODIFICATION DE L’HEURE	9
4.6. MODIFICATION DE L’OFFSET DE TEMPERATURE.....	9
4.7. EXEMPLE D’ENVOI D’UN PROGRAMME	9
5. ANNEXE A : LES ORDRES.....	10
6. ANNEXE B : LES OPTIONS SYSTEME.....	10
7. ANNEXE C : LES MODES D’APPARITION.....	11
8. ANNEXE D : LES OPTIONS D’AFFICHAGE	11
9. ANNEXE E : LES EFFETS.....	12
10. ANNEXE F: LES CODES D’ERREURS.....	12
11. ANNEXE G : STRUCTURES ET CONSTANTES	13
12. ANNEXE H : LA TABLE DE CARACTERES ASCII	13

1. Généralités

Le protocole peut être utilisé aussi bien sous une liaison RS232 que sous une liaison RS485, il est de type série, les paramètres de transmission sont les suivants :

- 1200 à 38400 bauds (9600 par défaut)
- aucune parité
- 8 bits de données
- 1 bit de stop

Les signaux DTR et DCD sont utilisés pour synchroniser l'émetteur (P.C., automate, calculateur...) et le récepteur (l'afficheur).

Connexion RS485 depuis un P.C. : Il est nécessaire d'utiliser le boîtier de conversion CV24 afin de convertir les signaux RS232 en signaux RS485.

1.1. Les Types de données stockés dans l'afficheur

Les données sont stockées sous la forme de fichiers dans la mémoire de l'afficheur, cette organisation est un peu semblable à celle d'un répertoire d'un disque dur de P.C.

1.1.1. Les données système

- Le logiciel d'actualisation de l'afficheur, -sauf afficheur D.I.-
- Polices de caractères, -sauf afficheur D.I.-
- Mot de passe, heure et date

1.1.2. Les données utilisateurs

- Les programmes ou messages
Un message est constitué du texte à afficher avec ses attributs (modes d'apparition, couleur...), et d'appels à des fichiers graphiques ou d'animations. Il peut être aussi constitué d'appels à d'autres messages avec leur date et leur heure d'exécution.
- Les graphiques et animations, -sauf afficheur D.I.-

1.1.3. Organisation de la mémoire

Tous les afficheurs sauf ceux de la série D.I possèdent deux types de mémoire, la mémoire conventionnelle et la Heap. La mémoire conventionnelle reçoit les programmes exceptés les fichiers graphiques et d'animations, sa taille se situe entre 32K et 64K selon le modèle d'afficheur. L'ordre EDITER permet d'écrire dans cette mémoire. La Heap (64Kb) contient les graphiques, les animations et les données système. L'ordre EDITFILE permet d'écrire dans la Heap.

La gestion de la Heap et donc des graphiques et des animations fait l'objet d'un autre document.

2. Le protocole de communication

2.1. Identification de l'afficheur:

L'afficheur a une adresse comprise entre 0 et 255 (0xFF). L'adresse par défaut est 1.

2.2. Constitution du message

La trame se compose des octets suivants :

SYN (0x16) (1 octet)	L'octet de synchronisation
NUM.B (2 octets)	Longueur du message (tous les octets de la trame)
N° (1 octet)	Adresse de l'afficheur
COD (1 octet)	Ordre à exécuter (voir annexe A)
DONNEE (N octets) Optionnel	Il peut s'agir du texte du programme, avec ses effets, ses modes d'apparition, etc., ou bien des données système
NOM.P (8 octets) Optionnel	Nom du programme (8 octets maximums). Les noms apparaissent en majuscules quand l'afficheur les reçoit, et seul est accepté l'ASCII de 7 bits.
CHEKSUM (2 octets)	Somme de tous les octets de la trame excepté le checksum
EditFile (5 octets)	Structure située après le nom du programme en EDITFILE qui indique le type de données envoyé ainsi que la taille du fichier. Description voir annexe G
GETFILE (8 octets)	Structure située après le nom du fichier en GETFILE. Indique la longueur et la position du paquet à transmettre. Description voir annexe G

Les paquets (trame) débutent par l'octet de synchronisation (SYN), puis deux octets qui indiquent la longueur du paquet. Le checksum est une somme de contrôle de 2 octets qui inclut tous les octets sauf les deux siens. Qu'il s'agisse de la longueur du paquet ou du checksum, l'octet de poids faible est transmis en premier.

L'afficheur confirmera toujours la réception d'un message par l'envoi de l'octet ACK (0x06) suivi d'un autre octet précisant s'il y a une erreur de transmission (voir annexe F).

Si l'afficheur reste silencieux et ne répond pas (pas de réponse avant deux secondes), on peut utiliser l'ordre GETNPAC pour vérifier si l'afficheur a tout de même reçu notre message.

Tous les ordres qui impliquent l'envoi de plus d'un octet de l'afficheur vers le DTE (par exemple le P.C.) seront transmis par l'intermédiaire de l'ordre ENVOYER, et avec le n° d'identification FEH.

2.3. Le Timing

Chaque octet doit être espacé du suivant d'au moins 1 ms. L'intervalle de temps entre le dernier octet d'un paquet qui parvient à l'afficheur et sa réponse varie entre 16 ms et 1 seconde. Les paquets sont donc séparés par au moins 16ms.

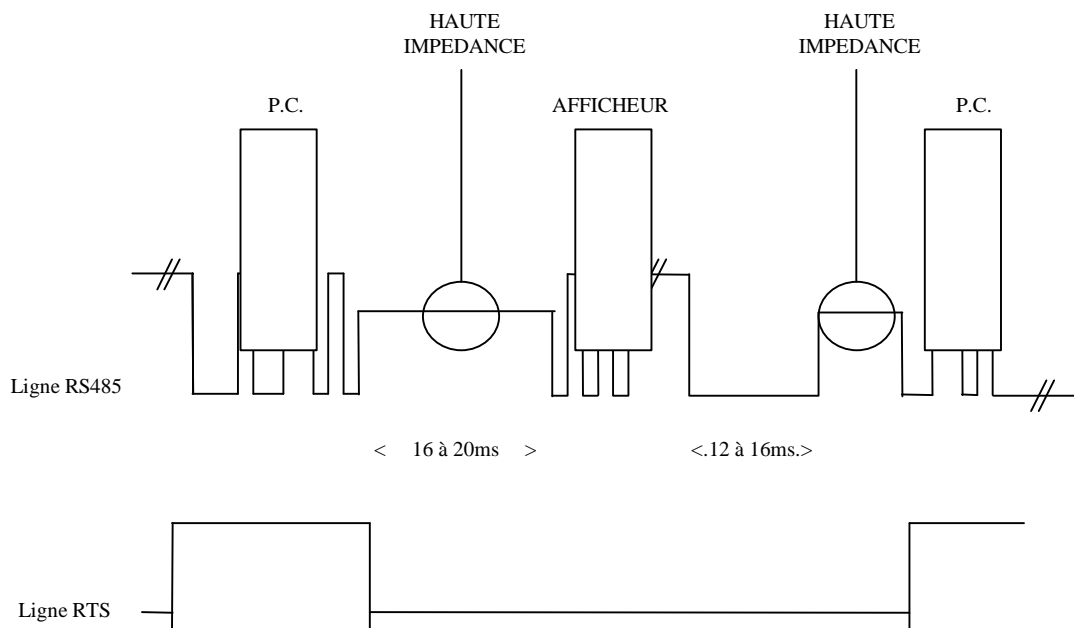
Si un afficheur reçoit mal un paquet, il est nécessaire d'attendre 100ms avant d'envoyer un nouveau message. Ce temps permet à l'afficheur de se préparer à la réception d'un nouveau paquet suite à une erreur de transmission.

Attention : si un octet est envoyé avec plus de 100 ms de retard par rapport au dernier octet envoyé, l'afficheur comprend que la transmission d'un autre paquet a commencé.

2.3.1. Le système RS485

Il est basé sur la transmission de données par l'intermédiaire de tensions différentielles. Il est conçu pour connecter sur une même ligne de communication jusqu'à 32 transmetteurs/récepteurs RS485.

Le système consiste en ce qu'il y ait seulement un transmetteur actif à un moment donné, tous les autres demeurent en réception et en état de haute impédance. Pour qu'il n'y ait pas de conflits de lignes, un récepteur peut se transformer en transmetteur uniquement s'il n'y a pas d'autre élément transmetteur.



Sur le graphique on montre comment doit s'établir la communication entre deux éléments d'une ligne RS485 et l'état de la ligne RTS. Cette dernière ligne doit être forcée à 1 lorsque le P.C. émet, puis à 0 pour que l'afficheur réponde.

Premièrement apparaît l'information transmise par le P.C., nous constatons que lorsque celle-ci se termine, le driver du P.C. se met en réception et demeure en haute impédance. Au bout de 16 à 20 millisecondes, l'afficheur commencera à transmettre si la réception a été correcte.

Remarque : dans le cas d'ordres, par exemple EDITER ou EFFACER, qui entraînent un long temps d'exécution (jusqu'à 1s) pour l'afficheur, la réponse de ce dernier est donc d'autant plus longue.

Une fois l'opération terminée, l'afficheur demeure en mode transmission pendant 12ms à 16ms. Puis de nouveau il passe en mode réception et on peut activer le driver du P.C. pour pouvoir transmettre un autre paquet. Si on ne respecte les temps, des conflits de lignes peuvent surgir et entraîner les erreurs de transmissions.

3. Création d'un programme

3.1. Généralité

Pour créer un message (programme) dans la mémoire de l'afficheur on utilise l'ordre EDITER, puis on utilise l'ordre ENVOYER pour composer le message, texte, ordre, effets, couleur..... et enfin l'ordre EXECUTER pour afficher le message.

Pendant que s'exécute un programme on ne peut en éditer un autre, l'ordre STOP permet d'arrêter un programme.

Par contre il est possible de composer un programme alors qu'un autre programme est en cours d'exécution. Il faut donc envoyer l'ordre EDITER du nouveau programme avant d'exécuter l'ancien programme.

Un programme est composé de 4 champs :

- Les options système
- Les modes d'apparition et le texte du message
- Les options d'affichage
- Les effets

3.2. Emission en plusieurs paquets

Il est conseillé d'utiliser des paquets d'au maximum 256 octets afin de limiter la probabilité d'erreurs de communication.

Les messages supérieurs à 256 octets devront être scindés en plusieurs paquets, ceci concerne uniquement l'ordre ENVOYER

Le compteur NUMPACK est mis à zéro quand on exécute l'ordre EDITER, on transmet alors les paquets ENVOYER les uns derrières les autres en vérifiant le ACK. Dans le cas où le ACK n'est pas renvoyé par l'afficheur, l'ordre GETNPACK nous permet de connaître le numéro du dernier paquet reçu correctement.

3.3. Les principaux ordres

L'ensemble des ordres sont récapitulés en annexe A

EDITER (COD = 0x06)

| SYN | NUM.B | N° | COD | NOM.PR | CHECKSUM |

Prépare le programme pour l'édition en le plaçant dans la zone d'édition. Si ce programme est déjà présent, le nouveau programme se substitue à l'ancien. S'il n'existe pas, on le crée.

D'autre part cet ordre met à zéro le compteur NUMPACK, utilisé par l'ordre ENVOYER lors de l'émission de paquets multiples.

ENVOYER (COD = 0x0C)

| SYN | NUM.B | N° | COD | DONNEES | CHECKSUM |

On utilise cet ordre pour envoyer des données à l'afficheur depuis le P.C. (et vice versa). Ce paquet peut contenir tout type de données : programme, polices ou logiciel. Ces données se placeront dans différents types de mémoire selon que l'on ait utilisé EDITER ou EDITFILE pour ouvrir l'archive.

EXECUTER (COD = 0x04)

| SYN | NUM.B | N° | COD | NOM.P | CHECKSUM |

GORGY TIMING

Protocole de communication pour gamme LEDI

EFFACER (COD = 0x05)

| SYN | NUM.B | N° | COD | NOM.P | CHECKSUM |

Efface un programme de la mémoire de l'afficheur.

STOP (COD = 0x03)

| SYN | NUM.B | N° | COD | CHECKSUM |

Arrête l'exécution d'un programme, s'il y en a un en cours d'exécution.

RESET (COD = 0x01)

| SYN | NUM.B | N° | COD | CHECKSUM |

Permet d'initialiser l'afficheur en effaçant tous les programmes en mémoire.

SETHEURE (COD = 0x0A)

| SYN | NUM.B | N° | COD | DATE+HEURE | CHECKSUM |

Programme le RTC (Real Time Clock) avec 6 octets qui indiquent respectivement l'année, le mois, le jour, l'heure, les minutes et les secondes.

GETHEURE (COD = 0x0B)

Demande l'heure à l'afficheur. L'afficheur renvoie un paquet avec l'ordre ENVOYER, le buffer possède l'heure dans le cinquième octet, les minutes dans le troisième octet, les secondes dans le premier octet, le jour dans le huitième octet, le mois dans le neuvième octet et l'année dans le dixième octet.

ATTENTION ! le format GETHEURE est différent de celui de SETHEURE

SETTEMP.(COD = 0x10) (cf. les exemples)

Modification de l'offset de température de l'afficheur.

GETTEMP.(COD = 0x11)

Demande la température et l'offset de l'afficheur. Celui-ci renvoie 2 octets avec l'ordre ENVOYER. La température se trouve dans le premier octet et l'offset dans le second octet (optionnel).

GETVER.(COD = 0x11)

| SYN | NUM.B | N° | COD | CHECKSUM |

Demande la version du logiciel

L'afficheur répond avec l'ordre ENVOYER

octet n°1 : la version du logiciel

octet n°2 : la version de la carte électronique

octets 3 à 6 : réservé

GETNPAC.(COD = 0x21)

| SYN | NUM.B | N° | COD | NOM.P. | CHECKSUM |

On utilise GETNPACK seulement quand un paquet ne renvoie pas ACK et que l'on veut s'assurer que la transmission est vraiment incorrecte. En effet il se peut que l'afficheur ne renvoie pas l'ACK bien qu'il ait reçu correctement le paquet.

L'afficheur renvoie le numéro du dernier paquet reçu correctement sous la forme suivante :

| ACK | NUMPACK |

CHECKSUM (COD : 0x07)

| SYN | NUM.B | N° | COD | CHECKSUM |

Demande le checksum du dernier paquet reçu. On doit recevoir le ACK l'octet haut du dernier checksum transmis.

NGETDIR (COD : 0x1E)

| SYN | NUM.B | N° | COD | CHECKSUM |

Demande à l'afficheur la structure du répertoire qui est constitué par des structures de type nom_arx. La fin du répertoire est déterminée par *pos vale NULL. La structure du répertoire est envoyée via le P.C (n°FEH) avec l'ordre ENVOYER.

GETFILE (COD : 0x20)

| SYN | NUM.B | N° | COD | NOM P. | GetFile | CHECKSUM |

Permet de récupérer dans l'afficheur n'importe quel fichier (en ram de programmes ou dans le heap) sous forme de paquets de taille variable. Ces derniers sont envoyés vers l'afficheur (destination = MASTER 0xFE) avec l'ordre ENVOYER. L'origine du paquet, étant donné la position initiale du fichier et sa longueur, est déterminée dans la structure Getfile qui se place dans le paquet qui suit le champ "NOM". Si dans la structure GetFile, le champ appelé pos est supérieur à la fin du paquet initial, l'afficheur indique l'erreur 22 (POS_TROP_GRAND).

PUTCNT (COD : 0x25)

| SYN | NUM.B | N° | COD | COMPTEUR | CHECKSUM |

Actualise un compteur déterminé dans un D.E.D.

FASTEXEC (COD : 0x27)

| SYN | NUM.B | N° | COD | DONNEE | CHECKSUM |

Transmet le contenu d'un programme qui s'exécutera de manière automatique sur l'afficheur, ce paquet permet une actualisation rapide des données sur l'afficheur.

3.4. Horodatage d'un message

L'option CYCLES permet d'horodater un message, il existe 3 formats

a . Horodatage par jour de la semaine : affichage du message certains jour de la semaine et à une certaine heure

CYCLES 0xC6**Jours de la semaine** *L, LM ou mVD etc...*

(L : Lundi, M : Mardi, m : Mercredi, J : Jeudi, V : vendredi, S : samedi, D :

Dimanche)

0x20**Heure de début** *09:00***0x20****Heure de fin** *12:30***0x2C****Nom du programme** *TEST*

b . Horodatage par date : affichage du message pendant un intervalle de temps

CYCLES 0xC6**Date de début** */01/08***0x20****Date de fin** */31/08***0x2C****Nom du programme** *TEST*

c . Horodatage par date et jour de la semaine : combinaison des deux premiers

CYCLES 0xC6**Date de début** */01/08***0x20****Date de fin** */31/08***0x20****Jours de la semaine** *L, LM ou mVD etc...*

(L : Lundi, M : Mardi, m : Mercredi, J : Jeudi, V : vendredi, S : samedi, D : Dimanche)

0x20**Heure de début** *09:00*

0x20
Heure de fin 12:30
0x2C
Nom du programme *TEST*

Remarque : les caractères en italique sont à convertir en code hexadécimal.

4. Exemples de transmissions :

Dans les exemples suivants, l'afficheur cible est à l'adresse 0x01.

4.1. Création d'un message

Tout d'abord il faut éditer un fichier, par exemple GORGY

a . Ordre EDITER, suivi du nom du fichier

16 0C 00 01 06 47 4F 52 47 59 B1 01

16H	: SYN.
000CH	Nombre d'octets (12)
01H	: N° afficheur (1)
06H	: Code de EDITER
474F524759	: Nom du fichier à éditer, 8 caractères ASCII (GORGY)
B101H	: Checksum (tous les octets ajoutés moins les deux du checksum)

Puis envoyer notre texte "AFFICHAGE TEST" dans le mode DEFILER

b . Ordre ENVOYER, suivi des données

16 16 00 01 0C E0 41 46 46 49 43 48 41 47 45 20 54 45 53 54 E7 04

16H	: SYN.
0016H	: Nombre d'octets (22)
01H	: N° afficheur (1)
0CH	: Code de ENVOYER
E04146464943	: Mode DEFILER puis le texte "AFFICHAGE TEST"
484147452054	
455354	
E704H	: Checksum (tous les octets ajoutés moins les deux du checksum)

Et enfin exécuter le programme GORGY.

GORGY TIMING

Protocole de communication pour gamme LEDI

c . Ordre EXECUTER n suivi du nom du programme

16 0C 00 01 1F 47 4F 52 47 59 01 CA

16H	: SYN.
000CH	: Nombre d'octets (12)
01H	: N° afficheur (1)
1FH	: Code de EXECUTER
474F524759	: Nom du programme (GORGY)
01CAH	: Checksum

4.2. Suppression d'un message

Suppression du programme GORGY
Ordre EFFACER, suivi du nom de l'archive que l'on veut effacer.

Par exemple :

16 0C 00 01 05 47 4F 52 47 59 B0 01

4.5. *Modification de l'heure*

Mise à l'heure de l'afficheur : 24 Octobre 1997 15h31 et 27s
 Ordre SETHEURE. suivi de l'année, mois, jour, heure, minute, seconde.

16 0D 00 01 0A 61 0A 18 0F 1F 1B FA 00

16H : SYN.
 000DH : Nombre d'octets (13)
 01H : N° afficheur (1)
 0AH : Code de SETHEURE
 610A180F1F1B : Année, mois, jour, heure, minute, seconde (97,10,22,15,31,27)
 00FAH : Checksum

4.6. *Modification de l'offset de température*

Ordre SETTEMP, suivi de l'offset de 10°C

16 08 00 01 10 0A 39 00

16H : SYN.
 0008H : Nombre d'octets (8)
 01H : N° afficheur (1)
 10H : Code de SETTEMP
 0AH : Offset de 10°C
 0039H : Checksum

4.7. *Exemple d'envoi d'un programme*

16 0B 00 01 06 54 45 53 54 68 01	EDITER 'TEST'
16 50 00 01	adresse 0x01, longueur 80 octets
0C C4 35 30	ENVOYER VIT_MODE 50
C1 30	TYPE_LETTRE 0
E0 41 46 46 49 43 48 41 47 45 20	DEFILER 'AFFICHAGE
4C 45 54 54 52 45 20 30	LETTRE 0'
C1 31	TYPE_LETTRE 1
E0 41 46 46 49 43 48 41 47 45 20	DEFILER 'AFFICHAGE
4C 45 54 54 52 45 20 31	LETTRE 1'
C1 32	TYPE_LETTRE 2
E0 41 46 46 49 43 48 41 47 45 20	DEFILER 'AFFICHAGE
4C 45 54 54 52 45 20 32	LETTRE 2'
F0 54 45 53 54	IMMEDIAT 'TEST'
F0 A7	IMMEDIAT HEURMIN
E6 18	Checksum
16 0B 00 01 1F 54 45 53 54 81 01	EXECUTER 'TEST'

5. ANNEXE A : LES ORDRES

ORDRES	CODES	EXPLICATIONS
RESET	01H	Reset de la mémoire
HARDRESET	02H	Reset de la carte électronique
STOP	03H	Arrête le programme en cours d'exécution
EXECUTER	1FH	Exécute un programme
EFFACER	05H	Efface un programme
EDITER	06H	Edite un programme
GETPROG	09H	Demande le code d'un programme
SETHEURE	0AH	Règle l'heure et la date de l'afficheur
GETHEURE	0BH	Demande la date et l'heure de l'afficheur
ENVOYER	0CH	Envoi des blocs de données
GETDIR	1EH	Demande le répertoire de l'afficheur, le dernier fichier renvoyé est celui en cours d'exécution
SETTEMP	10H	Règle l'offset de température
GETTEMP	11H	Demande l'offset et la valeur de la température
GETVER	12H	Demande la version du logiciel, type d'écran, etc...
GETNPAC	21H	Renvoie le numéro du dernier paquet reçu correctement

6. ANNEXE B : LES OPTIONS SYSTEME

GROSSEUR n	C0H	Epaisseur des lettres, chaque colonne occupera n colonne (1 à 4) de grosseur
TYPE_LETTRE n	C1H	Types de caractères (0 à 10 selon le modèle)
INVERSE	C2H	Inverse les couleurs
NORMAL	C3H	Annule INVERSE
VIT_MOD n	C4H	Vitesse de défilement (1 à 127)
PAUSE_MOD n	C5H	Temps d'attente entre les différents modes (0 à 2000 ms)
CYCLES	C6H	Horodatage : lancement programme à un instant t
LIGNE n	C7H	Numéro de la ligne
PROGRAMME	C8H	Exécute un programme dans un autre programme
SYNC	C9H	Synchronise les différentes lignes d'un afficheur multilignes
NO SYNC	CAH	Arrêt de la synchronisation des lignes
LANGUE n	CBH	Choix de la langue pour afficher la date 0 (Castillan) 1(Catalan) 2(Basque) 3(Galicien) 4(Français) 5(Anglais) 6(Portugais)
DATE_REF	CCH	Programmation de la date de référence (utilisé avec DIF_JOUR, DIF_SEM, DIF_MOIS)
ANIM	CEH	Exécute une animation
LUMIN	D0H	Règle la luminosité de l'afficheur de 1 à 100%

7. ANNEXE C : LES MODES D'APPARITION

DEFILER	E0H	Le texte se déplace de droite à gauche
CENTRE	E1H	Mode d'apparition
DIMINUER	E2H	Mode d'apparition
EMPILER	E3H	Mode d'apparition
TOURNER	E4H	Mode d'apparition
MONTER	E5H	Mode d'apparition
DESCENDRE	E6H	Mode d'apparition
ROTATION C	E7H	Mode d'apparition
ROTATION E	E8H	Mode d'apparition
ROTATION c	E9H	Mode d'apparition
ROTATION e	EAH	Mode d'apparition
APPARITION D	EBH	Mode d'apparition
APPARITION E	ECH	Mode d'apparition
APPARITION e	EDH	Mode d'apparition
APPARITION c	EEH	Mode d'apparition
NEIGE	EFH	Mode d'apparition
IMMEDIAT	F0H	Mode d'apparition
GLISSE	F1H	Mode d'apparition

8. ANNEXE D : LES OPTIONS d'AFFICHAGE

ANNEE	96H	Affiche l'année sur deux chiffres
MOIS	97H	Affiche le mois sur deux chiffres
NOM_MOIS	98H	Affiche le nom du mois en entier
JOUR	99H	Affiche le jour sur deux chiffres
NOM_JOUR	9AH	Affiche le nom du jour de la semaine en entier
HEURES	9BH	Affiche l'heure sur deux chiffres
MINUTES	9CH	Affiche les minutes sur deux chiffres
SECONDES	9DH	Affiche les secondes sur deux chiffres
DECIMAL	9EH	Affiche les dixièmes sur deux chiffres
TEMPERATURE	9FH	Affiche la température
CLIGNOTANT	A0H	Fait clignoter le texte situé entre deux CLIGNOTANT
COUL_TEXT	A1H	Couleur du texte
COUL_FOND	A2H	Couleur du fond
GRAPHIQUE n	A3H	Graphique à faire afficher
DIF_JOUR	A4H	Affiche la différence de jours avec DATE_REF
DIF_SEM	A5H	Affiche la différence de semaine avec DATE_REF
DIF_MOIS	A6H	Affiche la différence de mois avec DATE_REF
HEURMIN	A7H	Affiche l'heure et les minutes
TEMPER	A8H	Affiche la température + "°C"
JOUR_ABRE	A9H	Affiche le jour en abrégé
MOIS_ABRE	AAH	Affiche le mois en abrégé

9. ANNEXE E : LES EFFETS

CODES	CODES	EXPLICATIONS
CLIGNOTANT	B0H	- n clignotements sur l'afficheur
NEGATIF	B1H	- Inverse les couleur
EFFACER	B2H	- La ligne en cours d'affichage s'efface
ATTENTE n	B3H	- Attente n/4 secondes jusqu'au prochain effet

10. ANNEXE F: LES CODES D'ERREURS

| ACK | CODE ERREUR. |

Si l'ordre est arrivé correctement, le P.C. recevra cette confirmation. Elle possède un code de validation fixe (ACK) et un code d'erreur de l'afficheur.

Les valeurs des différents code d'erreur sont indiquées ci dessous

0	Aucun problème
1	Archive introuvable
2	Impossible d'éditer, il y a un programme en exécution
3	L'archive ne peut être créé
4	RAM pleine
5	Programme en exécution
6	Non permis pendant l'exécution d'un programme
7	Ordre inconnu
8	Archive vide
9	Mot de passe incorrect
A	Nom trop long
B	Format de date incorrect
C	Format de EDITFILE incorrect
D	La heap est pleine
E	Programme supérieur de FFF0H octets
F	Le programme ne peut s'exécuter
11	EDITFILE pas supporté
12	Configuration incorrecte
13	le checksum de la configuration n'est pas correcte

11. ANNEXE G : STRUCTURES ET CONSTANTES

STRUCTURE UTILISEE DANS L'ORDRE NGETDIR

Struct nom_arx{

```

        char*pos ;           */ Position écriture archive           */
        unsigned int size ; */ Taille de l'archive                */
        char nom [9] ;      */ Nom de l'archive terminé en 0      */
        char*posbuf ;       */ Position écriture EDITFILE        */
        unsigned int sizeact ;
        unsigned int sizebuf ; */ Taille du malloc EDITFILE        */
    };
    
```

12. ANNEXE H : LA TABLE DE CARACTERES ASCII

La plupart coïncident avec le code ASCII

CAR	COD	CAR	COD	CAR	COD	CAR	COD	CAR	COD	CAR	COD	CAR	COD
SP	20H	0	30H	°	40H	P	50H	ó	60H	p	70H	à	80H
!	21H	1	31H	A	41H	Q	51H	a	61H	q	71H	è	81H
"	22H	2	32H	B	42H	R	52H	b	62H	r	72H	ì	82H
Ñ	23H	3	33H	C	43H	S	53H	c	63H	s	73H	ò	83H
\$	24H	4	34H	D	44H	T	54H	d	64H	t	74H	ù	84H
%	25H	5	35H	E	45H	U	55H	e	65H	u	75H	ä	85H
&	26H	6	36H	F	46H	V	56H	f	66H	v	76H	ë	86H
'	27H	7	37H	G	47H	W	57H	g	67H	w	77H	ï	87H
(28H	8	38H	H	48H	X	58H	h	68H	x	78H	ö	88H
)	29H	9	39H	I	49H	Y	59H	i	69H	y	79H	ü	89H
*	2AH	:	3AH	J	4AH	Z	5AH	j	6AH	z	7AH	↑	8AH
+	2BH	;	3BH	K	4BH	Ç	5BH	k	6BH	é	7BH	ì	8BH
'	2CH	←	3CH	L	4CH	\	5CH	l	6CH	ú	7CH		
-	2DH	=	3DH	M	4DH	ç	5DH	m	6DH	í	7DH		
.	2EH	→	3EH	N	4EH	↓	5EH	n	6EH	ñ	7EH		
/	2FH	?	3FH	O	4FH	¿	5FH	o	6FH	á	7FH		



GORGY TIMING
LA MARQUE DU TEMPS

GORGY TIMING SA

8, Avenue Pierre de Coubertin Z.I Percevalière 7402.

38174 SEYSSINET CEDEX (Grenoble France).

Phone: +33 4 76 70 19 60 Fax: +33 4 76 49 06 21

e.mail : gorgy@gorgy-timing.com - www.gorgy-timing.com

PARIS agency Phone : +33 1 53 82 85 90 Fax: +33 1 53 82 06 22

Subsidiaries: Karlsruhe, **GERMANY**. Barcelona, **SPAIN**



Mark by EUROP UNION