



ADELE + MANUEL DE PROGRAMMATION



CHAPITRE 1 : DESCRIPTION GENERALE

1.1 Différences entre ADELE et ADELE+	4
1.2 Fonctionnement	4
La structure	4
1.2.1 La partie AXF	5
1.2.2 La partie PROGRAM	5
1.2.3 La partie DEFINITION	5
1.3 Téléchargement	5
1.4 Procédure de démarrage	7

CHAPITRE 2 : LES PARAMETRES DE COMMUNICATION

2.1 Description des paramètres de transmission	
2.1.1 Début de document	8
2.1.2 Début d' enregistrement	8
2.1.3 Fin d'enregistrement	8
2.1.4 Fin de document	9
2.1.5 Début de message d'erreur	9
2.1.6 Fin de message d'erreur	9
2.1.7 Caractère de remplissage de l'enregistrement	9
2.1.8 Nombre de caractère de synchronisation	9
2.1.9 Caractère de synchronisation	10
2.1.10 Longueur d'enregistrement	10
2.1.11 Vitesse de transmission	10
2.1.12 Configuration de la transmission	10
2.1.13 Envoi ou non du nombre de marques lues	11
2.1.14 Envoi ou non du nombre de marques horloges lues	11
2.1.15 Avertisseur sonore case bonne	11
2.1.16 Avertisseur sonore case rejet	11
2.1.17 Envoi ou non d'un message "prompt"	12
2.1.18 Envoi ou non des erreurs	12
2.1.19 Attente sur écho	12
2.1.20 Délai avant envoi des enregistrements	12
2.1.21 Délai entre caractères	13
2.1.22 Délai entre chaque enregistrement	13
2.1.23 Caractère de réémission de l'enregistrement	13
2.1.24 N° de programme	13
2.1.25 Vérification du nombre de marques horloges	13
2.1.26 Caractère pour lire un document	14
2.1.27 Caractère pour appeler l'enregistrement suivant	14
2.1.28 Caractère pour éjection case bonne	14
2.1.29 Caractère pour éjection case rejet	14
2.1.30 Distance avant arrêt papier (lecteur 926)	15
2.1.31 Caractère pour télécharger un programme	15
2.1.32 Caractère pour envoi de la version	15
2.1.33 Caractère pour impression interne (lecteur 995)	15
2.1.34 Saut du caractère de status	16
2.1.35 Longueur de l'enregistrement d'édition	16
2.2 Table des paramètres de communication	17

CHAPITRE 3 : LES INSTRUCTIONS

3.1 ACTIVE	18
3.2 BARCODE	19
3.3 CHECK	20
3.4 CLOCK	24
3.5 COPY	25
3.6 COUNTER	26
3.7 DISBUF	27
3.8 DISPLAY	28
3.9 EJECT	29
3.10 FIELD	30
3.11 FIELDXY	34
3.12 FILL	39
3.13 +GOTO et +GOSUB	40
3.14 +IF GOTO et +IF GOSUB	41
3.15 IFERROR	42
3.16 +IFMEMORY	45
3.17 INPUT	46
3.18 +LABEL	47
3.19 MEMORY	48
3.20 MESSAGE	50
3.21 NAME	52
3.22 NEXDOC	53
3.23 NEXREC	54
3.24 PRINT	56
3.25+ RETURN	57
3.26 RFIELD	58
3.27 RFIELDXY	59
3.28 SEND	60
3.29 SLEEP	61
3.30 TABLEA	62
3.31 TABLEM	63
3.32 TABLEN	64
3.33 TABLES	65
3.34 TABLET	67
3.35 WAITKEY	69
3.36 WRITEC	70
APPENDIX A : Liste des erreurs	72
APPENDIX B : Table ASCII	74
APPENDIX C : Glossaire	77
APPENDIX D : Conseil de programmation	79
APPENDIX E : Résumé des instructions et fonctionnalité	82
APPENDIX F : Exemple de programmation	89

CHAPITRE 1 : Description générale**1.1 DIFFERENCES ENTRE ADELE ET ADELE+**

ADELE permet de réaliser la lecture de programme "monodocument", par contre ADELE+ peut permettre de lire simultanément dans le même lot plusieurs types de documents mélangés.,ADELE ne disposant pas des fonctions de "saut" :

GOTO ,GOSUB ,IF GOTO ,IF GOSUB, RETURN ,LABEL , IFMEMORY

Ces fonctions seront repérées par un + dans le manuel

1.2 FONCTIONNEMENT

ADELE est un langage de programmation pour lecteur optique .ADELE doit être téléchargé à partir de l'ordinateur hôte vers le lecteur optique de marques (dit OMR) .Ce programme est sauvegardé en mémoire RAM non volatile .Il n'est donc pas nécessaire de retélécharger le programme tous les jours ,mais seulement en cas de modifications sur celui-ci.

ATTENTION : le lecteur peut ne pas voir certaines erreurs de syntaxe ,faites sous éditeur , Pour faciliter la mise en forme d'un programme ADELE ,il est conseillé d'utiliser le logiciel AXMPC ,qui en plus de son confort de travail ,présente l'avantage de contrôler les erreurs de syntaxe .

La structure d'un fichier source est la suivante :

@AXF	paramètres de communication
@PROGRAM	instructions
@DEFINITION	tables de décodages
@END	fin d'ADELE

Nota : les saut de ligne (LF) ,espaces ,et tabulations sont libres d'utilisation à tous moments

1.2.1 LA PARTIE "AXF "

Définie les paramètres de communication

l'ordre de ces paramètres ne doit pas être changé

les lignes de commentaires (partie gauche des lignes) doivent être en minuscules

1.2.2 LA PARTIE "PROGRAM"

Cette partie contient les instructions de lecture ,de contrôles ,de sous routine

seule une instruction par ligne est autorisé

les commentaires doivent être précédé du caractère ;

1.2.3 LA PARTIE "DEFINITION"

Cette partie contient les tables convertissant une marque lue en valeur ASCII

les commentaires doivent être précédé du caractère ;

aucun commentaire n'est possible à l'intérieur d'une table

1.3 TELECHARGEMENT D'UN PROGRAMME

Après la mise sous tension du lecteur ,envoi par l'ordinateur d' ESC (1BH) ,ou du caractère choisi dans la partie AXF (character to reload) .

Après 0,3 secondes ,le lecteur doit afficher :PARAMETRE SVP ou PARA .La même fonctionnalité peut être atteinte en appuyant ,sur le bouton de la face avant du lecteur quelques secondes,lorsqu'il affiche AUR .

Envoyer le fichier de paramétrage par la ligne V24 ,à la même vitesse que lors du dernier téléchargement .En cas d'incertitude ,après avoir envoyé l'ESC ,couper le lecteur et le rallumer après quelques secondes ;à ce moment le lecteur est revenu à 9600Bds ,pas de parité ,8bits ,1 stop .

Après être correctement téléchargé , le lecteur affiche OK ou L? ou ATTEND
COMMANDE

Cette page est laissée intentionnellement blanche

1.4 PROCEDURE DE DEMARRAGE

- ❶ Allumer le lecteur, il doit afficher "L" ou "DOCUMENT SVP" . Le lecteur est prêt à fonctionner avec les paramètres définis par le dernier téléchargement .
- ❷ Avec la la commande "V" , le lecteur envoie le numéro de série d'ADELE .
- ❸ Avec "L" (en accord avec les paramètres) , le lecteur lit le premier document .
- ❹ Le lecteur envoie les données et attend une décision , soit "G" ou "S" (pour bon ou mauvais) .

CHAPITRE 2 : les paramètres de communication

Les paramètres de communication permettent de créer l'enregistrement de lecture au format requis .

Le fichier de paramétrage doit commencer par le caractère utilisé pour le retéléchargement (par défaut <ESC>) . Il est possible de placer des lignes de commentaires entre ce caractère et la section @AXF , mais ces commentaires ne doivent pas contenir le caractère utilisé pour la demande de version , par défaut V. La chaîne @AXF indique le début du fichier des paramètres de communication . L'ordre des paramètres , et le nombre de données *ne doivent pas être changés* . Les caractères compris entre 30 H et 45 H ne sont pas autorisés .

2.1 Description des paramètres de communication**2.1.1 caractères de début de document**

Cette fonction permet de définir jusqu'à 3 caractères à envoyer avant chaque document .

Exemple :

caractères de début de document	01 FF FF = <i>SOH</i>
---------------------------------	-----------------------

2.1.2 caractères de début d'enregistrement

Cette fonction permet de définir jusqu'à 3 caractères à envoyer avant chaque enregistrement

Exemple :

caractères de début d'enregistrement	02 FF FF = <i>STX</i>
--------------------------------------	-----------------------

2.1.3 caractères de fin d'enregistrement

Cette fonction permet de définir jusqu'à 3 caractères à envoyer après chaque enregistrement

Exemple :

caractères de fin d'enregistrement	0D 0A 03 FF = <i>CR LF ETX</i>
------------------------------------	--------------------------------

☞ Les commentaires en italique ne doivent pas apparaître dans le fichier de paramétrage.

2.1.4 caractère de fin de document

Cette fonction permet de définir jusqu'à 3 caractères qui marqueront la fin du document .

Exemple :

caractères de fin de document	04 FF FF = <i>EOT</i>
-------------------------------	-----------------------

2.1.5 caractère de début des messages d'erreur

Cette fonction permet de définir jusqu'à 3 caractères qui seront envoyés au début des messages d'erreur .

Exemple :

caractères de début de message d'erreur	15 FF FF = <i>NACK</i>
---	------------------------

2.1.6 caractère de fin des messages d'erreur

Cette fonction permet de définir jusqu'à 3 caractères qui seront envoyés à la fin des messages d'erreur .

Exemple :

caractères de fin de message d'erreur	0D 0A 03 = <i>CR LF ETX</i>
---------------------------------------	-----------------------------

2.1.7 caractère de remplissage de l'enregistrement

Si l'enregistrement est de longueur variable , la fin de l'enregistrement peut être complétée par ce caractère .

Exemple :

caractère de remplissage	2E = <i>."</i>
--------------------------	----------------

2.1.8 nombre de caractères d'entête

Cette fonction permet de définir un nombre X de caractères Y (voir 2.1.9) au début de l'enregistrement .Ces caractères seront inclus dans la longueur de l'enregistrement .

Exemple :

quantité de caractères de synchro	00
-----------------------------------	----

👉 Les commentaires en italique ne doivent pas apparaître dans le fichier de paramétrage.

2.1.9 caractère de synchronisation

Valeur ASCII du caractère choisi.

Exemple :

caractère de synchro	2D = "-"
----------------------	----------

2.1.10 longueur de l'enregistrement

La longueur de l'enregistrement peut être variable (dans ce cas 00) ou fixe .La longueur de l'enregistrement doit être codée en binaire.

Si la valeur est différente de 0 chaque enregistrement sera tronqué à cette valeur et le dernier enregistrement sera complété , si nécessaire . Cette valeur peut varier de 0000 à FFFF .

Exemple :

longueur d'enregistrement	lsb hex	28
longueur d'enregistrement	msb hex	00

Dans ce cas ,l'enregistrement est de 40 caractères .

2.1.11 baud rate

Le débit binaire peut être choisi ici comme suit :

1 = 1200 , 2 = 2400 , 3 = 4800 , 4 = 9600 , 5 = 19200 , 6 = 38400

Exemple :

vitesse de transmission (baudrate)	04 = 9600
------------------------------------	-----------

2.1.12 uart

Le mode de fonctionnement de l'UART est défini ici :

bits de données	02 = 7 bits	03 = 8 bits
avec parité	00 = pas de parité	01 = avec parité
choix de parité	00 = impaire	01 = paire
stops bits	01 = 1 stop	03 = 2 stops

Exemple :

longueur data (sept ou huit)	03
validation parité	00
parité paire ou impaire	01
nombre de stop bits (un ou deux)	01

👉 Les commentaires en italique ne doivent pas apparaître dans le fichier de paramétrage.

2.1.13 envoi du nombre de marques lues

Le lecteur peut envoyer le nombre de marques lues sur le document , ce nombre est sur 4 caractères .

00 = pas d'envoi

01 = envoi du nombre de marque

Exemple :

envoi du nb marques lues (oui,non)	00 = <i>pas d'envoi</i>
------------------------------------	-------------------------

2.1.14 envoi du nombre de marques horloges lues

Le lecteur peut aussi envoyer le nombre de marques horloges lues sur un document , ce nombre est sur 3 caractères .

00 = pas d'envoi

01 = envoi du nombre de marque
horloge

Exemple :

envoi du nb d'horloges lues (oui ,non)	00 = <i>pas d'envoi</i>
--	-------------------------

2.1.15 Avertisseur sonore, document bon

Cette option vous permet de sélectionner un signal sonore émis pour chaque document bon.

00 = rien

01 = son très court

02 = son court

03 = son moyen

04 = son "normal"

05 = alarme

06 = son modulé

Exemple :

son case bonne	00 = <i>pas de son</i>
----------------	------------------------

2.1.16 Avertisseur sonore,document mauvais.

Cette option permet d'émettre un son après une erreur, avec les mêmes valeurs que ci dessus .

Exemple :

son case rejet	05 = <i>alarme</i>
----------------	--------------------

👉 Les commentaires en italique ne doivent pas apparaitre dans le fichier de paramétrage.

2.1.17 envoi de message d'initialisation

Cette fonction valide ou dévalide l'envoi d'un message à la mise sous tension .

00 = pas de message 01 = message

Exemple :

envoi prompt à la mise secteur (oui,non)	00 = <i>pas de message</i>
--	----------------------------

2.1.18 envoi des erreurs

Lorsqu'une erreur apparaît (status), il est possible de l'envoyer sur l' hôte ou de ne pas l'envoyer .

00 = pas d'envoi 01 = envoi 1<x<FF envoi retardé du magasin vide FF pas de retour magasin vide

Exemple :

envoi des messages d'erreur (oui,non)	01 = <i>envoi d'erreur</i>
---------------------------------------	----------------------------

La réponse FF à cette question ,permet de demander l'envoi de tous les messages d'erreur (status) ,sur l'ordre d'appel du document ; excepté celui de magasin vide (code 06) . Dans ce cas l'ordre d'appel du document est enregistré ,et le lecteur restera en attente indéfiniment d'un document à lire .

La réponse avec une valeur x (en secondes) ,demandera l'envoi de tous les messages excepté celui du magasin vide ,mais ici le lecteur ne restera en attente de document que pendant le temps x ;passé celui-ci le lecteur enverra le code 06 .

2.1.19 attente d'écho

Si 01 est sélectionné dans cette option ,le lecteur , après chaque envoi d 'un caractère, attend en retour le même caractère .

00 = pas d'écho 01 = avec écho

Exemple :

attente d'un écho (oui,non)	00 = <i>pas d'écho</i>
-----------------------------	------------------------

2.1.20 délai avant l'envoi des données

La valeur définie ici donne le délai d'attente avant l'envoi des données d'un document , en millisecondes (de 0 à FF) .

Exemple :

délai avant envoi données (cinquante ms)	0A = 10, <i>pas de 50 millise.</i>
--	------------------------------------

2.1.21 délai après chaque caractère

La valeur définie ici donne le délai d'attente avant l'envoi d'un caractère .

Exemple :

délai après chaque caractère	<i>0C = 12 milliseconde</i>
------------------------------	-----------------------------

☞ les commentaires en italique ne doivent pas apparaître dans le fichier de paramétrage

2.1.22 délai après chaque enregistrement

La valeur définie ici donne le délai d'attente avant l'envoi de chaque enregistrement .

Exemple :

délai après chaque enregist.(une ms)	<i>1F = 31 millisecondes</i>
--------------------------------------	------------------------------

2.1.23 caractère de réémission

Le caractère défini ici est celui utilisé pour renvoyer le contenu du dernier document .

Exemple :

caractère pour réémission enregistrement	<i>52 = R character</i>
--	-------------------------

2.1.24 numéro de programme

Ce paramètre n'est pas valide avec ADELE.

2.1.25 controle des marques horloges

Si cette valeur est égale à zéro, le nombre de marques horloges n'est pas vérifié. Si cette valeur est supérieure à 0 , le contrôle s'effectue sur cette valeur .

Exemple :

vérification marques horloges <i>= 40</i>	<i>28 = test si le nombre de marque horloge</i>
--	---

2.1.26 caractère pour appeler un document

Le lecteur attend le caractère défini ici pour initialiser le cycle de lecture . Si la valeur est 0 , le lecteur lit en continu .

Exemple

caractère pour appeler un document	4C = L
------------------------------------	--------

☞ Les commentaires en italique ne doivent pas apparaître dans le fichier de paramétrage.

2.1.27 caractère pour appeler un nouvel enregistrement .

Le lecteur attend le caractère défini ici pour envoyer le nouvel enregistrement ;si sa valeur est 0 ,le lecteur envoie les enregistrements à la suite .

Exemple :

caractère pour appeler enregist. suivant	00 = <i>pas d'attente</i>
--	---------------------------

2.1.28 caractère de éjection case bonne du document

Le lecteur attend le caractère défini ici pour éjecter un document en case bonne . Si cette valeur est 00 ,le lecteur éjecte automatiquement le document en accord avec les tests définis par le programme ADELE .

Exemple :

caractère pour envoi case bonne	47 = <i>G</i>
---------------------------------	---------------

2.1.29 caractère pour rejeter le document

Le lecteur attend le caractère défini ici pour rejeter un document en case "rejet".

Exemple :

caractère pour envoi case rejet	53 = <i>S</i>
---------------------------------	---------------

2.1.30 distance avant l'arrêt du papier

Pour pouvoir lire une étiquette code à barre ,certains lecteurs nécessitent un arrêt de la feuille plus loin sous la tête . La valeur du pas est de 0,5 mm par unité (de 0 à 255) .

Exemple :

distance avant arrêt papier	$14 = 20Steps = 10\text{ mm}$
-----------------------------	-------------------------------

☞ Les commentaires en italique ne doivent pas apparaître dans le fichier de paramétrage.

2.1.31 Caractère pour retélécharger les paramètres

Ce paramètre permet à l'utilisateur de retélécharger de nouveaux paramètres à n'importe quel moment , sans avoir à presser le bouton pendant l'initialisation de la mise sous tension .Si ce caractère est reçu lorsque le lecteur est en attente de document , le lecteur acceptera les nouveaux paramètres. Par défaut cette valeur est 1B .

Exemple:

caractère pour retélécharger	$1B = ESC$
------------------------------	------------

2.1.32 Envoi du numéro de version

Ce paramètre définit le caractère à envoyer pour recevoir le numéro de version du firmware du lecteur . Ce caractère doit etre envoyé lorsque le lecteur est en attente de document .Ce caractère permet aussi de recevoir le nom et la taille du programme déjà téléchargé dans le lecteur .

Exemple:

caractère pour demande de version	$56 = V$ <i>cette valeur est fixe</i>
-----------------------------------	---------------------------------------

2.1.33 Caractère pour la commande d'impression

Ce paramètre n'est pas applicable sous ADELE ,avec les lecteurs actuels .

☞ Les commentaires en italique ne doivent pas apparaître dans le fichier de paramétrage.

2.1.34 Saut du caractère de status

Ce caractère voir fonctions (MESSAGE ,IFERROR ,CHECK) ,est envoyé en tout début de l'enregistrement ,transmis vers l'ordinateur, contenant les données lues .

Exemple :

Saut du caractère de status	non	<i>envoi du status</i>
-----------------------------	-----	------------------------

2.1.35 Longueur de l'enregistrement d'édition

L'enregistrement d'édition (à ne pas confondre avec celui de transmission) peut avoir une longueur de 1 à 12000 caractères .Il est possible ici d'en limiter sa taille , permettant ainsi de reduire les temps de lecture et de dégager de la mémoire utile .

Exemple :

Longueur de l'enregistrement d'édition	2000	<i>2000 car d'édition</i>
--	------	---------------------------

👉 Les commentaires en italique ne doivent pas apparaitre dans le fichier de paramétrage.

2.2 Tableau des paramètres de communication

 $@_{AXF}$

caractères de début de document		01 FF FF FF	ASCII
caractères de début d'enregistrement		02 FF FF FF	"
caractères de fin d'enregistrement		0D 0A 03 FF	"
caractères de fin de document		04 FF FF FF	"
caractères de début de message d'erreur		15 FF FF FF	"
caractères de fin de message d'erreur		0D 0A 03 FF	"
caractère de remplissage		2E	
quantité de caractères de synchro		00	0 à 255 (binaire)
caractère de synchro		2D	ASCII
longueur d'enregistrement	lsb hex	28	0 for variable length
longueur d'enregistrement	msb hex	00	
vitesse de transmission (baudrate)		04	
uart			
longueur data sept ou huit bits		03	02 = 7 bits 03 = 8 bits
validation parité		00	00 = no parity 01 = parity on
parité paire ou impaire		01	00 = odd 01 = even
nombre de stop bits	un ou deux	01	01 = 1 stop 03 = 2 stops
envoi du nb de marques lues	oui non	00	00 = pas d'envoi 01 = envoi # marques
envoi du nb d'horloges lues	oui non	00	00 = pas d'envoi 01 = envoi # clocks
son case bonne		00	00,01,02,03,04,05,06
son case rejet		05	00,01,02,03,04,05,06
envoi prompt à la mise secteur	oui non	00	00 = no prompt, 01 = prompt
envoi des messages d'erreur	oui non	01	00 = pas d'envoi 01 = envoi 1 = délai 1 à FF délai infini
attente d'un écho	oui non	00	00 = pas d'attente 01 = attend écho
délai avant envoi données (cinquante ms)		00	00 to 255 (binaire)
délai après chaque caractère (une ms)		00	00 to 255 (binaire)
délai après chaque enregist.(une ms)		00	00 to 255 (binaire)
caractère pour réémission enregistrement		52	ASCII, 00 = pas d'attente
numéro de programme		00	non significatif for ADELE
vérification marques horloges		00	00 = pas de verif., >00 = rejet
<i>si</i>			<i>pas égal</i>
caractère pour appeler un document		4C	ASCII, 00 = pas d'attente
caractère pour appeler enregist. suivant		00	ASCII, 00 = pas d'attente
caractère pour envoi case bonne		47	ASCII, 00 = pas d'attente
caractère pour envoi case rejet		53	ASCII
distance avant arrêt papier		00	00 to 255(1 unité = 0.5 mm)
caractère pour retélécharger		1B	ASCII 01 to FF
caractère pour demande de version		56	forcé par ADELE à V
caractère demande d'impression interne		50	00 pas de commande d'impression
saut du caractère de status		0	envoi du status 1 pas d'envoi
longueur de l'enregistrement d'édition		12000	de 1 à 12000

CHAPITRE 3 : Les instructions**3.1 ACTIVE**

Cette instruction permet à l'utilisateur de définir une séquence de caractères autorisant la lecture.

Cela permet ,lorsque le lecteur est connecté à un ordinateur hôte ,d'ignorer tout flux d'information non précédé d'un entête.

ACTIVE "psw"

psw n'importe quelle chaîne de caractères ,jusqu'à 18 caractères, terminée par un CR ,et cadré par des " "

exemple: ACTIVE "Sepsi"

ATTENTION :

Après l'utilisation de l'instruction ACTIVE , le lecteur entre en mode verrouillé . Il est possible de télécharger un autre paramètre par l'envoi de la chaîne définie avec l'instruction ACTIVE.

3.2 BARCODE

Cette instruction permet la lecture d'un code à barre .

BARCODE dest nbr format

dest:	destination dans l'enregistrement	0,1 a 12000
nbr:	numéro du code à barre (ADELE est multiétiquettes)	1 a 99
format:	format de sortie	P ou N

- Si la destination est 0 , le résultat de la lecture ira à la position indiquée dans la MEMORY 54 ,typiquement l'adresse la plus haute déjà utilisée dans l'enregistrement d'édition .
- Nbr donne le numéro du code à barre ,par exemple si 3 étiquettes sont lues sur le document et seule la deuxième est utilisée , l'instruction BARCODE xxx 2 y le permet .
- Une instruction BARCODE doit être utilisée pour chaque étiquette.

Format N ou P

N envoie seulement le contenu du code à barre

P envoie des délimiteurs:[] encadrant :

un premier caractère indiquant le type de code lu :

A	=	codabar du premier caractère au dernier
B	=	codabar du dernier caractère au premier
E	=	code 39 du premier caractère au dernier
F	=	code 39 du dernier caractère au premier
I	=	2/5 du premier au dernier
J	=	2/5 du dernier au premier
T	=	EAN 8 ou 13 du premier au dernier
S	=	EAN 8 ou 13 du dernier au premier

et un chiffre après le code indiquant la distance de l'étiquette jusqu'à la fin du document

exemple :[A1238]

Exemple :

BARCODE 10 1N

3.3 CHECK

Cette instruction permet de faire des contrôles de modulus (chiffre clef) sur des portions de l'enregistrement .

CHECK	MODULO	début	long.	BUFFER	tab	sortie	action
	MODULO : 3 calculs possibles			MODULO10			
				MODULO11			
				MODULO10REC			
début :		Adresse ,dans l'enregistrement,de début du contrôle					1 à 12000
long. :		Nombre de caractère à partir de "début" pris pour le contrôle					1 à
BUFFER :		Optionnel ; si présent le texte pris pour "message d'erreur" ,le sera dans enregistrement d'édition ,à partir de l'adresse contenu variable "tab"					
dans la							
tab :		N° de la table où prendre le message d'erreur					1 à 199
		ou position dans l'enregistrement					1 à 12000
sortie :		Où envoyer le message d'erreur					SERIAL,NO,PRINTER
action :		Que doit faire le lecteur après une erreur					C,I,M,N,K

MODULO = type de contrôle à effectuer

MODULO10 c'est le contrôle effectué dans la numérotation EAN
les multiplicateurs sont alternativement 3 et 1 en partant de la droite vers la gauche
(en sautant le dernier caractère qui est le chiffre de contrôle)

ex :	zone à contrôler	7	6	0	1	2	3	4	1
	multiplicateurs		3	1	3	1	3	1	3
	résultat = 49	21 +	6 +	0 +	1 +	6 +	3 +	12	

Pour obtenir le modulo on divise se résultat (49) par 10 ;et on soustrait à 10 le reste de cette division (ici reste = 9 ;10 - 9 = 1) ,le chiffre obtenu est le modulo que le lecteur doit trouver dans l'enregistrement juste après le dernier chiffre de la zone de contrôle (ici le numéro devait donc être 76012341)

MODULO10RECursif : se calcul d'après une table de la gauche vers la droite

MODULO11

les multiplicateurs vont de 2 à 7 ,puis repartent à 2 ,de la droite à la gauche (en sautant le dernier caractère qui est le chiffre de contrôle)

ex :	zone à contrôler	7	6	0	1	2	3	4	2
	multiplicateurs		2	7	6	5	4	3	2
	résultat = 86	14 +	42 +	0 +	5 +	8 +	9 +	8	

Pour obtenir le modulo on divise le résultat (86) par 11 ,et on soustrait à 11 le reste de la division (ici reste = 9 ,11 - 9 = 2) le modulo est de 2

Début = adresse de départ de la zone à contrôler ,dans l'enregistrement d'édition

Si l'adresse de ce paramètre est 0 l' adresse de départ sera pris dans la MEMORY 57

Long. = Nombre de caractères

Nombre de caractères ,à partir de "début", à prendre dans l'enregistrement d'édition pour faire le contrôle .Cette longueur doit comprendre la position du modulo .

BUFFER : option

Si cette option est présente le texte utilisé pour le message d'erreur sera pris dans l'enregistrement d'édition (buffer) ,à partir de l'adresse définie dans le paramètre "tab".La fin du message devra avoir été matérialisé dans l'enregistrement par le caractère Héxa FF ,ceci par l'instruction FILL . En cas d'absence de cette option le paramètre "tab" indiquera la table message (TABLEM) ,où il faut prendre le message à envoyer .

Tab = N° de table où position dans l'enregistrement

est Voir explication BUFFER .Si l'option BUFFER est absente et ce paramètre "tab" à 0 ,le lecteur ira chercher le message d'erreur dans celle dont le numéro sera contenu dans la MEMORY 59.Si la mémoire est aussi à 0 ,il n'y aura pas d'envoi du message d'erreur

sortie = où le message sera envoyé

SERIAL : sortie sur la ligne V24 (RS232)
 NO : pas de message envoyé
 PRINTER : le message sera envoyé sur une interface Centronics
 (si le lecteur est équipé de l'option imprimante)

action = Ce que doit faire le lecteur après détection d'une erreur

- Si ADELE détecte une erreur ,il est possible de définir dans quel bac le document sera éjecté , ce que devra faire l'opérateur ;ainsi que de l'existence ou non d'un "drapeau" d'erreur
- Le drapeau d'erreur est le premier caractère de l'enregistrement . Sa valeur détermine si une erreur a été détectée sur le document , et si c'est le dernier enregistrement de ce document .

Les valeurs pour action peuvent être :

	status d'erreur	envoi données	bac	envoi message	touche	Mémoire 62
IGNORE		oui	bon	non	non	
KEY	oui	oui	rejet	oui	oui	1
CONTINUE	oui	oui	rejet	oui	non	2
MARK	oui	oui	bon	oui	non	4
NOSEND	oui	non	rejet	oui	oui	8
ABORT *	oui	non	rejet	oui	non	16
SENDTEXT	non	oui	bon	oui	non	

Les valeurs dans la mémoire 62 peuvent s'additionner ,il est de même possible d'écrire sur cette mémoire pour changer "l'action"

Les valeurs pour le drapeau d'erreur

<i>valeur du drapeau</i>	<i>erreur sur le document</i>	<i>dernier enregistrement</i>
0	non	non
1	oui	non
2	non	oui
3	oui	oui

Exemple : CHECK MODULO10 156 8 BUFFER 55 SERIAL CONTINUE

cet exemple va faire un contrôle de modulo 10 à partir de l'adresse 156 ,dans l'enregistrement d'édition , sur une longueur de 8 caractères (7 chiffre de base + 1 qui est le modulo) .Le message d'erreur sera pris dans l'enregistrement d'édition à partir de l'adresse 55 jusqu'au caractère FF .Ce message sera transmis vers l'ordinateur via la V24 .Le document sera envoyé dans la case rejet sans arrêt du lecteur (si le lecteur possède une case de rejet ,sinon rejet avec arrêt "Erreur logique") .

Exemple : CHECK MODULO10 156 8 55 SERIAL CONTINUE

cet exemple va faire un contrôle de modulo 10 à partir de l'adresse 156 ,dans l'enregistrement d'édition , sur une longueur de 8 caractères (7 chiffre de base + 1 qui est le modulo) .Le message d'erreur sera pris dans la TABLEM (table message) N° 55 .Ce message sera transmis vers l'ordinateur via la V24 .Le document sera envoyé dans la case rejet sans arrêt du lecteur (si le lecteur possède une case de rejet, sinon rejet avec arrêt "Erreur logique") .

3.4 CLOCK

Cette instruction permet d'écrire le nombre de marques horloges lues, dans l'enregistrement d'édition .

CLOCK dest face

dest : adresse de l'enregistrement où le nombre de marques horloges sera copié. Si égal à 0, le nombre de marques horloges sera écrit juste après la dernière adresse utilisée par les données précédentes .

face : 1 pour le recto, 2 pour le verso .Le résultat est toujours écrit sur 3 caractères

Exemple :

CLOCK 120 1

La lecture d' un document de 66 marques horloges sur la face 1, place "066" à la position 120 de l'enregistrement d'édition .

3.5 COPY

Cette instruction permet de copier X caractères depuis une position dans l'enregistrement à un autre endroit dans ce même enregistrement d'édition .

COPYFROM début TO dest LENGTH long

début : adresse du premier caractère à copier. Si 0 est utilisé
 le contenu de la MEMORY 58 sert comme index.

dest : adresse où sera copié la chaîne.

long : nombre de caractère à copier.

Exemple:

COPY FROM 10 TO 25 LENGTH 3

copiera 3 caractères à partir de l'adresse 10 dans l'adresse 25 .

3.6 COUNTER

Cette instruction est utilisée pour identifier les groupes dans un champ. Par exemple , sur un champ contenant 50 articles ,en utilisant counter et FE dans l'option "Si pas marqu" de l'instruction FIELD , seuls les articles marqués seront envoyés avec leur numéro d'ordre par rapport à la valeur définie dans cette instruction. (le 1er produit ayant le N° de cette valeur)

Dans l'instruction FIELD le paramètre "compteur" détermine la longueur de ce compteur.

COUNTER valeur

valeur : de 0 à 65500

Exemple :

COUNTER 301

FIELD 4 1 1 16 5 1 **FE** HOR DOWN **3** 0 1 0 1

le compteur commence à 301 sur une longueur de 3 .

3.7 DISBUF

Cette instruction permet d'envoyer les 16 caractères ,à partir de l'adresse sélectionnée, sur l'afficheur du lecteur (4 caractères sur les petits afficheurs) .

DISBUF adresse

adresse : adresse dans le buffer d'édition à partir de laquelle seront extraits les 16 caractères à envoyer au lecteur .Les caractères non affichables seront remplacés par des espaces .Les minuscules seront remplacées par des majuscules.

Note : si le paramètre adresse est 0 ,l'adresse prise , pour aller déterminer le point de départ dans l'enregistrement d'édition , sera la valeur contenue dans la mémoire 54 (MEMORY 54) .

Exemple :

DISBUF 1023

enverra le texte contenu dans l'enregistrement d'édition ,de l'adresse 1023 à l'adresse 1038 incluse , sur l'afficheur du lecteur

Note : il est conseillé de coupler cette instruction avec une instruction PAUSE ,permettant ainsi aux opérateurs d'avoir le temps de lire ce texte .Il y sinon risque d'affichage trop rapide d'un autre message .

3.8 DISPLAY

Cette instruction permet d'envoyer un message sur l'afficheur du lecteur.

DISPLAY texte

texte: texte alphanumérique .Si le texte est plus grand que la longueur de l'afficheur il sera coupé sur sa droite .L'AXM 926 accepte 4 caractères , les AXM 952 ,985 , 995 acceptent 16 caractères.

Exemple :

DISPLAY SEPSI

enverra le texte "SEPSI" sur l'afficheur du lecteur

Note : il est conseillé de coupler cette instruction avec une instruction PAUSE ,permettant ainsi aux opérateurs d'avoir le temps de lire ce texte .Il y sinon risque d'affichage trop rapide d'un autre message .

3.9 EJECT

Cette instruction sera utile en cas de correction en ligne (dite ON LINE) , c'est à dire où, en cas de rejet d'un document pour mauvaise lecture ,l'on souhaite corriger immédiatement au clavier de l'ordinateur la zone erronée .Dans ce cas il est utile d'éjecter le document pour permettre une lecture de celui-ci ,sans attendre l'ordre d'éjection de l'ordinateur .

EJECT case

case : case du lecteur où doit être éjecter le document

GOOD ,BAD

GOOD = case bonne
BAD = case mauvaise

Note : certains lecteurs n'ont pas de case mauvaise ,il convient donc sur ceux-ci de ne faire que des éjections GOOD .

3.10 FIELD

Définit comment interpréter un champ .Le résultat de cette interprétation sera ajouté au contenu de l'enregistrement à l' adresse définie dans les paramètres . Cette instruction est valable pour le recto comme pour le verso du document.La différence entre le recto et le verso se fait par le numéro de colonne (1 à 48 pour le recto, 51 à 98 pour le verso)

FIELD dest,nbcar,ligne,col,group,saut,tab,pasmar,dg,df,compt,min,max,grlong,fitem

dest :	index de la position de la première donnée dans l'enregistrement	0,1 à 12000
nbcar :	nombre de caractères à mettre dans l'enregistrement pour chaque marque lue	1 à la longueur choisie
ligne :	numéro de la ligne supérieure du champ	1 à 119
col :	numéro de la colonne la plus à gauche du champ	1 à 48
group :	nombre de groupes dans un champ	1 à 48 ou 119
saut :	nombre de lignes ou de colonnes à sauter entre deux groupes	1 à 48 ou 119
tab :	numéro de la table de décodage	1 à 199
pasmar :	valeur héra du caractère à écrire si aucune marque dans le groupe	00 à FF H
dg :	direction du groupe	VER ou HOR
dc :	direction du champ	LEFT,RIGHT,UP,DOWN
compteur :	longueur du compteur	0 à 5
min :	nombre minimum de marques par groupe	0 à X
max. :	nombre maximum de marques par groupe	0 à X
grlong :	nombre d'éléments utilisés dans la table pour un groupe	0 à Y
fitem :	N° de la position de l'élément de la table servant de 1ère valeur pour le groupe	1 à Y

Nota : X est la longueur du groupe ;Y la longueur de la table .

La table définie chaque marque, de la 1ère à la dernière,incluant les marques sautées.

dest = position de la première donnée dans l'enregistrement

- La donnée calculée après conversion ,doit être placée dans l'enregistrement.Avec ce paramètre ,on indique où cette donnée doit apparaître. L'enregistrement peut faire jusqu'à 12000 caractères . Si la destination est 0 ,la donnée sera placée après la dernière écrite dans l'enregistrement.

nbcar = nombre de caractères à mettre dans l'enregistrement

- Le nombre de caractères pour représenter une marque ,dans un groupe ,doit être défini ici.
- Cette valeur va de 1 à 5 pour une TABLEN(table numérique) ,est de 1 pour une TABLEA (table ASCII),et pour la TABLES (table chaîne) de 1 à la longueur de la chaîne.

ligne = numéro de la ligne supérieure du champ

- Ce nombre donne la position de référence du champ.Cette position est comptée depuis la première marque horloge du document .Il est défini de 1 à 119.

col = numéro de la colonne la plus a gauche du champ

- Ce numéro donne la position de référence du champ.C'est toujours la colonne la plus à gauche du champ. Ce numéro peut varier de 1 à 48 pour le côté recto et de 51 à 98 pour le côté verso.

group = nombre de groupe dans un champ

- Un champ est composé d'un ou de plusieurs groupes ayants les mêmes valeurs de décodification . On défini ici le nombre de groupes appartenants à un champ . Cette valeur peut varier de 1 à 48 pour un groupe vertical et de 1 à 119 pour un groupe horizontal.

saut = nombre de groupe à sauter

- Si les groupes ne sont pas contigus ,le pas du saut entre chaque groupe doit être indiqué ici. .
- Cette valeur peut varier de 1 à 48 (vertical) ou 119 (horizontal) .

tab = identification du numéro de la table de décodage

- Chaque groupe est décodé en accord avec une table de conversion. On indique ici le numéro de la table correspondant aux valeurs à donner aux marques du groupe
- Un numéro de table varie entre 1 et 199 .

pasmar = valeur utilisée en cas de non marquage

- S'il n'y a pas de marques dans un groupe ,la valeur définie ici sera écrite dans l'enregistrement d'édition .Cette valeur doit être donnée en Hexadécimal (ex : 20 pour un espace) .Cette instruction peut être utilisée aussi lorsque l'on veut laisser la position dans l'enregistrement libre en cas de non marquage.Par exemple, si une seule marque est autorisée , mais sur plus d'un groupe , et que seul le groupe marqué doit générer un caractère . Dans ce cas le paramètre doit être FF pour incrémenter le pointeur d'enregistrement , ou FE pour laisser le pointeur à la même place (ceci est utilisé pour les données compactées)

dg = direction du groupe

- Un groupe se définit sur une colonne ou une ligne . Il peut avoir une direction de gauche à droite ou de bas en haut .
- Les paramètres pour dg peuvent être HOR ou VER.

dc = direction du champ

- Un champ est composé d'un ou de plusieurs groupes . Ce paramètre, définit ici dans quelle direction se repètent les groupes qui forment le champ ,depuis son origine. Ces directions se définissent avec le document marques horloges sur le côté droit . Pour un champ avec un seul groupe ce paramètre n'agit pas.
- Les valeurs de ce paramètre peuvent être:

LEFT	si le second groupe est sur le côté gauche
RIGHT	si le second groupe est sur le côté droit
UP	si le second groupe est vers le haut
DOWN	si le second groupe est vers le bas

compteur = nombre de caractères utilisés par le compteur de groupe

- Chaque groupe dans un champ est compté . La valeur de début de ce compteur est donné par l' instruction COUNTER .Ce paramètre défini si le compteur doit apparaître dans l' enregistrement et si oui sur combien de chiffres.
- Ce paramètre varie de 0 (pas d'écriture) à 5 .

min = nombre minimum de marques dans un groupe

- Dans chaque groupe, le nombre de marques lues sera comparé avec cette valeur . Si ce nombre est inférieur, une erreur sera générée (voir l' instruction IFERROR) .
- Ce paramètre varie de 0 jusqu'à la longueur du groupe .

max = nombre maximum de marques dans un groupe

- Dans chaque groupe, le nombre de marques lues sera comparé avec cette valeur . Si ce nombre est inférieur une erreur sera générée (voir l' instruction IFERROR) .
- Ce paramètre varie de 0 jusqu'à la longueur du groupe .

longr = longueur du groupe

- Ce paramètre indique le nombre d'items à utiliser dans la table choisie . Cela permet d'utiliser une même table pour plusieurs groupes ;en utilisant 0 ,on utilise toute la table .

fitem = premier item utilisé dans la table

- Ce paramètre donne l'item de départ à utiliser dans la table choisie . Si "longr" est à 0 ,Fitem doit être a 1.

- Exemple :

La TABLEN 0,1,2,3,4,5,6,7,8,9,0 peut être utilisée pour un champ contenant les marques 0,1,2,3 tout aussi bien que pour un champ avec les marques 1,2,3,4,5,6,7,8,9,0 ;dans le premier cas ,longr est à 4 et fitem à 1 ; dans le second cas , longr est 0 et fitem à 1 .

3.11 FIELDXY

Définit comment interpréter un groupe réparti sur plusieurs sous groupes .Le résultat de cette interprétation sera ajouté au contenu de l'enregistrement à l' adresse définie dans les paramètres .

Cette instruction est valable pour le recto comme pour le verso du document.La différence entre le recto et le verso se fait par le numéro de colonne (1 à 48 pour le recto, 51 à 98 pour le verso).

Cette fonction autorise des sous groupe horizontaux ou verticaux .

La table utilisée doit contenir la définition individuelle de chaque alvéole en commençant à l'origine du champs (en haut à gauche du premier groupe) .Le nombre d'éléments de cette table sera égal au nombre de sous groupes multiplié par la longueur d'un groupe (= donc le nombre total d'alvéole du groupe) .

Le groupe doit être de forme rectangulaire ,mais il est possible de jouer sur les caractères de saut dans la table pour obtenir une forme quelconque .

On ne peut utiliser que des tables : TABLEA , TABLEN , TABLES

L'ensemble des sous groupes constituant le groupe ,les tests de plausibilité se feront donc sur l'ensemble des alvéoles .

FIELDXY dest,nbcar,ligne,col,group,saut,tab,pasmar,dg,df,compt,min,max,grlong , fitem

dest :	index de la position de la première donnée dans l'enregistrement	0,1 à 12000
nbcar :	nombre de caractères à mettre dans l'enregistrement pour chaque marque lue	1 à la longueur choisie
ligne :	numéro de la ligne supérieure du champ	1 à 119
col :	numéro de la colonne la plus à gauche du champ	1 à 48
group :	nombre de groupes dans un champ	1 à 48 ou 119
saut :	nombre de lignes ou de colonnes à sauter entre deux groupes	1 à 48 ou 119
tab :	numéro de la table de décodage	1 à 199

pasmar :	valeur héxa du caractère à écrire si aucune marque dans le groupe	00 à FF H
dg :	direction du groupe	VER ou HOR
dc :	direction du champ	LEFT,RIGHT,UP,DOWN
compteur :	longueur du compteur	0 à 5
min :	nombre minimum de marques par groupe	0 à X
max. :	nombre maximum de marques par groupe	0 à X
grlong :	nombre d'éléments utilisés dans la table pour un groupe	0 à Y
fitem :	N° de la position de l'élément de la table servant de 1ère valeur pour le groupe	1 à Y

Nota : X est la longueur du groupe ;Y la longueur de la table .

La table définit chaque marque, de la 1ère à la dernière,incluant les marques sautées.

dest = position de la première donnée dans l'enregistrement

- La donnée calculée après conversion ,doit être placée dans l'enregistrement.Avec ce paramètre ,on indique où cette donnée doit apparaître. L'enregistrement peut faire jusqu'à 12000 caractères . Si la destination est 0 ,la donnée sera placée après la dernière écrite dans l'enregistrement.Ne pas oublier que FIELDXY peut générer dans l'enregistrement plusieurs écritures ,résultat de la multiplication de *nbcар* (nombre de caractères par alvéole) par *max* (nombre maximum de marques pour le groupe)

nbcар = nombre de caractères à mettre dans l'enregistrement

- Le nombre de caractères pour représenter une marque ,dans un groupe ,doit être défini ici.

- Cette valeur va de 1 à 5 pour une TABLEN(table numérique) ,est de 1 pour une TABLEA (table ASCII),et pour la TABLES (table chaîne) de 1 à la longueur de la chaîne.

ligne = numéro de la ligne supérieure du champ

- Ce nombre donne la position de référence du champ. Cette position est comptée depuis la première marque horloge du document . Il est défini de 1 à 119.

col = numéro de la colonne la plus à gauche du champ

- Ce numéro donne la position de référence du champ. C'est toujours la colonne la plus à gauche du champ. Ce numéro peut varier de 1 à 48 pour le côté recto et de 51 à 98 pour le côté verso.

group = nombre de groupe dans un champ

- Un champ est composé d'un ou de plusieurs groupes ayant les mêmes valeurs de décodification . On définit ici le nombre de groupes appartenant à un champ . Cette valeur peut varier de 1 à 48 pour un groupe vertical et de 1 à 119 pour un groupe horizontal.

saut = nombre de groupe à sauter

- Si les groupes ne sont pas contigus , le pas du saut entre chaque groupe doit être indiqué ici .
- Cette valeur peut varier de 1 à 48 (vertical) ou 119 (horizontal) .

tab = identification du numéro de la table de décodage

- Chaque groupe est décodé en accord avec une table de conversion. On indique ici le numéro de la table correspondant aux valeurs à donner aux marques du groupe
- Un numéro de table varie entre 1 et 199 .

pasmar = valeur utilisée en cas de non marquage

- non utilisé dans cette instruction , si une alvéole est non cochée elle est simplement ignorée et rien n'est écrit dans l'enregistrement d'édition .

dg = direction du groupe

- Un groupe se définit sur une colonne ou une ligne . Il peut avoir une direction de gauche à droite ou de bas en haut .
- Les paramètres pour dg peuvent être HOR ou VER.

dc = direction du champ

- Un champ est composé d'un ou de plusieurs groupes . Ce paramètre, définit ici dans quelle direction se repètent les groupes qui forment le champ ,depuis son origine. Ces directions se définissent avec le document marques horloges sur le côté droit . Pour un champ avec un seul groupe ce paramètre n'agit pas.
- Les valeurs de ce paramètre peuvent être:

LEFT	si le second groupe est sur le côté gauche
RIGHT	si le second groupe est sur le côté droit
UP	si le second groupe est vers le haut
DOWN	si le second groupe est vers le bas

compteur = nombre de caractères utilisés par le compteur de groupe

- ce paramètre n'est pas utilisé dans cette instruction ,**le laisser à 0**

min = nombre minimum de marques dans un groupe

- Dans chaque groupe, le nombre de marques lues sera comparé avec cette valeur . Si ce nombre est inférieur, une erreur sera générée (voir l' instruction IFERROR) .
- Ce paramètre varie de 0 jusqu'à la longueur du groupe .

max = nombre maximum de marques dans un groupe

- Dans chaque groupe, le nombre de marques lues sera comparé avec cette valeur . Si ce nombre est inférieur une erreur sera générée (voir l' instruction IFERROR) .
- Ce paramètre varie de 0 jusqu'à la longueur du groupe .

longr = longueur du groupe

- Ce paramètre indique le nombre d'items à utiliser dans la table choisie . Cela permet d'utiliser une même table pour plusieurs groupes ;en utilisant 0 ,on utilise toute la table .

fitem = premier item utilisé dans la table

- Ce paramètre donne l'item de départ à utiliser dans la table choisie . Si "longr" est à 0 ,Fitem doit être a 1.
- Exemple :

La TABLEN 0,1,2,3,4,5,6,7,8,9,0 peut être utilisée pour un champ contenant les marques 0,1,2,3 tout aussi bien que pour un champ avec les marques 1,2,3,4,5,6,7,8,9,0, ;dans le premier cas ,longr est à 4 et fitem à 1 ; dans le second cas , longr est 0 et fitem à 1 .

*Cette page à été laissée blanche
intentionnellement*

3.12 FILL

L'enregistrement d'édition reste inchangé après la lecture du document . Il convient de le "nettoyer" entre chaque document . On peut utiliser entre chaque document lu une partie commune , qui peut être lue sur un premier document et que l'on souhaite conserver sur les feuilles suivantes . Pour ce faire ,il faut alors nettoyer que la partie non commune

Le caractère choisi pour remplir l'enregistrement doit être le même que celui choisi dans AXF au paramètre "character used to fill the record" **si** on utilise les options automatiques des instructions FIELD ou SEND .

FILL FROM aa TO bb WITH car

aa	adresse de départ de remplissage de l'enregistrement	1 à 12000
bb	adresse de fin de remplissage de l'enregistrement	1 à 12000
car	caractère en Héra choisi pour le remplissage	20H à

FFH

Exemple :

FILL FROM 100 TO 1000 WITH 41

Remplissage de l'enregistrement à partir de 100 jusqu'à 1000 avec le caractère A .

3.13 GOTO et GOSUB (+)

Ces instructions permettent un saut inconditionnel à un label défini . Ce label doit être défini dans le programme .

L' instruction GOSUB attend un RETURN à la fin de la sous routine . Il est possible d'imbriquer jusqu'à 20 sous routines .

GOTO label
GOSUB label

label	numéro du label où s'effectue le saut	1 à 199
-------	---------------------------------------	---------

Exemple:

```
GOTO 12  
GOSUB 43
```


3.14 IF GOTO et IF GOSUB (+)

Cette commande compare deux morceaux de l'enregistrement d'édition et exécute un saut si le résultat est vrai . L'instruction IF GOSUB attend un RETURN en fin de sous routine .

IF () aa	opérateur	() bb LENGTH lon	GOTO label
IF () aa	opérateur	() bb LENGTH lon	GOSUB label
aa		adresse de la première variable du test, si 0 l'index sera le contenu de la mémoire 60 .	0,1 à 12000
opérateur		nature du test	> < = #
bb		adresse de la deuxième variable du test, si 0 l'index sera le contenu de la mémoire 61 .	0,1 à 12000
lon		longueur de la chaîne à tester	1 à 255
label		numéro du label où aura lieu le saut si le résultat du test est vrai	1 a 255

Exemple :

```
IF ( ) 10 # ( ) 20 LENGTH 5 GOTO 3
```

- 1) comparaison des 5 caractères commençant à l'adresse 10 de l'enregistrement avec les 5 caractères commençant à l'adresse 20 .
- 2a) si les deux chaînes sont différentes aller au label 3
- 2b) si les chaînes sont égales continuer le programme

3.15 IFERROR

Cette instruction permet de définir ce qu'il faut faire après qu'ADELE ait détecté une erreur. Cette instruction peut apparaître n'importe où, dans le programme et restera valide jusqu'à une nouvelle instruction IFERROR (généralement une instruction IFERROR est placée juste avant une instruction FIELD pour définir sur ce champ, les tests, les messages et l'action du lecteur, en cas d'erreur de lecture).

IFERROR BUFFER ,texte,sortie,valeur,deja,soit,lgrp,aumn,tous,seul,action

	BUFFER	option : message d'erreur dans l'enregistrement	présent/absent
	texte	numéro de la table contenant le message d'erreur	0 à 199
	sortie	où le message doit être envoyé	SERIAL,NO,PRINTER.
	valeur	valeur à mettre dans l'enregistrement en cas d'erreur	0 à
F0			
	déjà	test si l'enregistrement est déjà écrit à cette adresse	0 ou 1
	soit	test si le groupe est tout marqué ou tout vide	0 ou 1
	lgrp	test à partir du premier groupe marqué tous marqués	0 ou 1
	aumn	au moins un groupe doit être marqué dans le champ	0
ou 1			
	tous	chaque groupe doit être correctement marqué	0 ou 1
	seul	test si un et seulement un groupe est marqué	0
ou 1			
	action	que doit faire le lecteur après une erreur	C,I,M,N

BUFFER = message dans l'enregistrement d'édition

- si cette option est présente, indique au programme que le libellé du message d'erreur, doit être pris dans l'enregistrement d'édition, et non pas dans une table message (TABLEM). La section de l'enregistrement désirée comme message, doit se terminer par un caractère FF ; ce caractère aura été préalablement écrit par une instruction FILL.

texte = numéro de la table, ou adresse enregistrement, contenant le message d'erreur

- Après la détection d'une erreur, un texte peut être envoyé sur une sortie. Ce message est programmé soit dans une TABLEM, dont la valeur peut varier de 1 à 199 ; soit dans une portion de l'enregistrement d'édition, dans ce cas la valeur peut être de 1 12000.

Si cette position est à 0, le lecteur ira chercher le N° de table dans la mémoire 59. Si le contenu de cette mémoire est lui aussi à 0, il n'y aura pas d'envoi de message d'erreur. Cette mémoire est remise à zéro à chaque mise sous tension.

sortie = où le message sera envoyé

SERIAL	: sortie sur la ligne V24 (RS232)
NO	: pas de message envoyé
PRINTER	: le message sera envoyé sur une interface Centronics (si le lecteur est équipé de l'option imprimante)

valeur = valeur envoyée en cas d'erreur sur un groupe ou un champ

- Ce paramètre définit le code héxa qui sera envoyé dans l'enregistrement de sortie à la place des données définies dans la table correspondante ,en cas de mauvaises lectures .

- Il existe 6 possibilités de test qui peuvent être levées (0 ou 1)

déjà = test si l'enregistrement est déjà écrit

- Ce test est utilisé lorsque deux CHAMPS ou plus ont été définis pour écrire leur résultat à la même adresse dans l'enregistrement d'édition .

soit = test si tous les groupes sont marqués ou vides

- Ce test permet de vérifier si un champ est totalement marqué. ou totalement vide . Une erreur est reportée dans le cas contraire .

1grp = test de marquage du premier groupe marqué , jusqu'au dernier

- A partir du premier groupe marqué dans un champ tous les autres groupes doivent être marqués . Ce test envoie les données sans contrôle des zéros non significatifs

aumn = test qu'au moins un groupe doit être marqué dans le champ

tous = test si chaque groupe est correctement marqué

seul = test si un et seulement un groupe est marqué

action = Ce que doit faire le lecteur après détection d'une erreur

- Si ADELE détecte une erreur ,il est possible de définir dans quel bac le document sera éjecté , ce que devra faire l'opérateur ;ainsi que de l'existence ou non d'un "drapeau" d'erreur
- Le drapeau d'erreur est le premier caractère de l'enregistrement . Sa valeur détermine si une erreur a été détectée sur le document , et si c'est le dernier enregistrement de ce document .

Les valeurs pour action peuvent être :

	status d'erreur	envoi données	bac	envoi message	touche	Mémoire 62
IGNORE		oui	bon	non	non	
KEY	oui	oui	rejet	oui	oui	1
CONTINUE	oui	oui	rejet	oui	non	2
MARK	oui	oui	bon	oui	non	4
NOSEND	oui	non	rejet	oui	oui	8
ABORT *	oui	non	rejet	oui	non	16
SENDTEXT	non	oui	bon	oui	non	

Les valeurs dans la mémoire 62 peuvent s'additionner ,il est de même possible d'écrire sur cette mémoire pour changer "l'action"

Les valeurs pour le drapeau d'erreur

<i>valeur du drapeau</i>	<i>erreur sur le document</i>	<i>dernier enregistrement</i>
0	non	non
1	oui	non
2	non	oui
3	oui	oui

Note : il est impératif d'avoir ,en début de programme ,un IFERROR avant toute instruction de lecture (BARCODE ,FIELD) .

3.16 IFMEMORY (+)

Cette instruction permet de faire des tests sur les mémoires internes du lecteur
Pour la définition des mémoires voir l'instruction MEMORY .

IFMEMORY n opérateur const GOTO label
IFMEMORY n opérateur const GOSUB label

IFMEMORY n opérateur MEMORY n2 GOTO label
IFMEMORY n opérateur MEMORY n2 GOTO label

n	numéro d'identification de la mémoire	1 à 149
opérateur	opérateur du test	< > = #
const	constante utilisée par l'opérateur arithmétique	1 à 149
n2	numéro de la seconde mémoire	1 à 149
label	numéro du label de branchement si le test est vrai	1 à 149

Exemple :

IFMEMORY 101 > MEMORY 125 GOSUB 121

signifie : Si le contenu de la mémoire 101 est plus grand que le contenu
de la mémoire 125 , ADELE sautera au label 121 avant de
continuer le programme .

3.18 LABEL (+)

Un label est une instruction atteinte dans le programme par un GOTO ou un GOSUB .

LABEL idnbr

idnbr est un numéro entre 1 et 199

Les labels ne peuvent être des caractères alphanumériques , mais uniquement des chiffres.

exemple :

LABEL 12

3.19 MEMORY

Cette instruction permet d'effectuer différentes opérations sur les mémoires internes. Elles sont de 3 catégories :

- Les mémoires ADELE : elles permettent de passer ou d'obtenir des informations de la programmation ADELE .
- Les mémoires OMR : elles permettent de passer ou d'obtenir des informations des drivers du lecteur (ex : nombre de marques horloges lues) .
- Les mémoires utilisateur : elles permettent pour l'utilisateur d'effectuer des boucles de programme, des calculs , des comparaisons de valeur ,etc....

Les opérations sur les mémoires peuvent s'effectuer par rapport à des constantes ou à d'autres mémoires .

MEMORY n op const
MEMORY n op MEMORY n2
MEMORY n tra adr LENGTH long

n	est le numéro de mémoire	1 à 149
op	est l'opérateur du calcul	+, -, =
const	est la constante utilisée pour le calcul	
n2	est le numéro de la seconde mémoire	1 à 149
tra	est l'opérateur de transfert	FROM, TO
adr	est l'adresse dans l'enregistrement d'édition	(0) , 1 à 12000
long	est le nombre de caractère à transférer sur TO ou FROM	(1 à 5)

Si l'opérateur de transfert est TO et l'adresse de destination 0 , le contenu de la mémoire 54 sera utilisé comme adresse de destination .

Les mémoires sont prédéfinies ainsi :

De 1 à 19 pour les valeurs ADELE :

- | | | | |
|----|---------------------------|----|--------------------------------|
| 1 | adresse de destination | 2 | nombre de caractère par groupe |
| 3 | position de la ligne | 4 | position de la colonne |
| 5 | nombre de groupes | 6 | saut pour le prochain groupe |
| 7 | N° de table "données" | 8 | caractère si pas de marque(s) |
| 9 | direction du groupe | 10 | direction du champs |
| 11 | nb digit du compteur | 12 | nb min de marque |
| 13 | nb max de marque | 14 | longueur du groupe |
| 15 | 1er élément dans la table | | |

Note : La valeur de la mémoire 8 se donne en décimal et non en hexadécimal

De 50 à 70 pour les valeurs du lecteur :

	50	nb de documents "bons" lus	51	nb de documents "erronés" lus
	52	nb total de documents lus	53	numéro du record
*	54	adresse par défaut dans l'enregistrement	55	nb de marques horloges recto
	56	nb marques horloges verso	57	index pour start de CHECK
	58	index source pour COPY	59	index pour MESSAGE et IFERROR
	60	1er paramètre d'index pour IF	61	2ème index pour IF
	62	valeur du flag d'erreur (instructions CHECK, IFERROR, MESSAGE)		

* **Cette mémoire est automatiquement mise à jour par les instructions qui utilisent l'index d'adressage dans l'enregistrement ,mais seulement si le nouvel index est supérieur au plus haut déjà utilisé .**

**Les mémoires 50 à 53 sont remises à zéro après chaque téléchargement
La mémoire 59 est remise à zéro à chaque mise sous tension du lecteur**

De 100 à 149 libres à l'utilisateur (valeurs comprises entre 0 et 65535 autorisées)

application :

FIELD	101	5	y.x.	1	:	met la mémoire 54 à 106
FIELD	50	3	y.x.	1	:	ne bouge pas la mémoire 54
COPY	FROM	10	to	0	LENGTH	3 met la mémoire 54 à 109

Exemple 1 : MEMORY 7 + 1
pour la prochaine instruction RFIELD ,la table suivante sera utilisée.

Exemple 2 : MEMORY 1 = 0
pour la prochaine instruction RFIELD ,les données seront écrites dans l'enregistrement d'édition à la prochaine adresse libre (adresse par défaut mémoire 54).

Exemple 3 : MEMORY 4 - 4
pour la prochaine instruction RFIELD ,la colonne d'origine du champ sera 4 colonnes plus près des marques horloges.

Exemple 4 : MEMORY 101 = MEMORY 50
copie le nombre de documents lus "bons" dans la mémoire 101

Exemple 5 : MEMORY 101 FROM 1243 LENGTH 2
prends 2 caractères à partir de l'adresse 1243 de l'enregistrement d'édition et le met dans la mémoire 101.

Exemple 6 : MEMORY 103 + MEMORY 105
la mémoire 103 recevra la somme de la mémoire 103 plus la valeur de la mémoire 105.

3.20 MESSAGE

Cette instruction est utilisée pour envoyer un message vers un périphérique après avoir effectué tout type de test .

MESSAGE BUFFER tab device action

BUFFER	option : message d'erreur dans l'enregistrement d'édition	absent présent
tab	est le numéro de la table qui contient le message à envoyer	0 à 199
sortie	est le type de périphérique où doit être envoyé le message	SERIAL PRINTER NO
action	donne ce que doit faire le lecteur lorsque l'erreur est détectée	C I K M N

BUFFER = message dans l'enregistrement d'édition

- si cette option est présente, indique au programme que le libellé du message , doit être pris dans l'enregistrement d'édition, et non pas dans une table message (TABLEM) .La section de l'enregistrement désirée comme message , doit se terminer par un caractère FF ; ce caractère aura été préalablement écrit par une instruction FILL .

tab = numéro de la table , ou adresse enregistrement ,contenant le message

- Ce message est programmé soit dans une TABLEM ,dont la valeur peut varier de 1 à 199 ; soit dans une portion de l'enregistrement d'édition ,dans ce cas la valeur peut être de 1 12000.

Si cette position est à 0 ,le lecteur ira chercher le N° de table dans la mémoire 59 .Si le contenu de cette mémoire est lui aussi à 0 ,il n'y aura pas d'envoi de message d'erreur . Cette mémoire est remise à zéro à chaque mise sous tension .

Sortie = Type de périphérique où le message doit être envoyé
les messages d'erreur doivent être envoyés sur un périphérique ;mais ce type n'est pas forcément le même pour tous les messages .

SERIAL	: est la sortie série (V24,RS232) standard
PRINTER	: est la sortie optionnelle de type centronics du lecteur
NO	: le message ne sera pas transmis

Action = comment doit se comporter le lecteur lors de la détection de l'erreur
 -ce paramètre va donc permettre de définir dans quelle case ranger les documents (rejet ou non),ainsi que l'action que devra faire l'opérateur .
 -de plus ,un "flag" d'erreur sera mis en 1ère position de l'enregistrement transmis à la lecture ;il permettra de savoir si une erreur à été détectée et si c'est le dernier enregistrement de la feuille lue (il est possible de transmettre le contenu d'un seul document en plusieurs enregistrements).

Les valeurs pour le paramètre action sont :

	status d'erreur	envoi données	bac	envoi message	touche	Mémoire 62
IGNORE		oui	bon	non	non	
KEY	oui	oui	rejet	oui	oui	1
CONTINUE	oui	oui	rejet	oui	non	2
MARK	oui	oui	bon	oui	non	4
NOSEND	oui	non	rejet	oui	oui	8
ABORT *	oui	non	rejet	oui	non	16
SENDTEXT	non	oui	bon	oui	non	

* Le traitement du document est stoppé lorsque cette erreur est détectée

Les valeurs de la mémoire 62 peuvent s'additionner ,de même il est possible d'écrire sur cette mémoire pour changer "l'action"

Les valeurs pour le "flag" d'erreur sont :

Valeur du "flag"	erreur sur le document	dernier enregistrement
0	non	non
1	oui	non
2	non	oui
3	oui	oui

Exemple

MESSAGE 101 SERIAL MARK

signifie : si le programme passe à cette ligne ,il enverra le texte défini dans la table 101 vers la sortie série ,le document aura son "status" en erreur ,mais sera rangé dans le bac des "bons"

Note : en cas d'utilisation ,à l'intérieur de l'instruction MESSAGE ,des options ABORT ou NOSEND ,il est quant même obligatoire ,de passer ensuite par les instructions SEND et NEXDOC .Il n'y aura alors pas de transmission de record ,mais ces instructions donnent l'ordre de rejet au lecteur (NEXDOC) ,et ferme un record vide pour le host pour permettre de le libérer et ainsi de renvoyer un nouvel ordre de lecture (SEND) .

3.21 NAME

Cette instruction permet à la mise sous tension d'afficher le nom du programme en mémoire .Cet affichage permet donc à l'opérateur d'être sûr que c'est le bon programme qui est en mémoire du lecteur ,évitant ainsi de retélécharger ,dans le doute , avant chaque lancement d'une lecture .Ce nom sera aussi envoyé dans le message "status" que peut demander l'ordinateur sur lequel est connecté le lecteur ,dans ce cas la procédure de contrôle peut être automatique (contrôle du nom du programme et retéléchargement si nécessaire) ; d'autant plus que le lecteur, dans le status, envoie aussi la quantité de mémoire restant disponible ,permettant, ainsi, de vérifier l'intégrité de ce programme .

NAME "**nom**"

nom donne le nom à affecter au programme ,ce nom doit être saisi entre ""

Exemple : NAME "Lecture QCM"

à la mise sous tension (après les auto-tests) l'écran du lecteur affichera

LECTURE QCM

Note : les caractères non affichables seront remplacés par des blancs , les minuscules par des majuscules

3.22 NEXDOC

Cette instruction doit finir tout programme ,elle donne en effet l'ordre d'appel du document suivant (next document) .Elle suit donc généralement une instruction SEND (transmission de la lecture) .Par habitude on placera ces deux instruction (SEND et NEXDOC) dans un LABEL (ex 199 qui est le plus haut possible) ,et chaque fin de lecture d'un document passera par ce label (ex par un GOTO) .Ne pas oublier ,que même après une option ABORT sur un MESSAGE il est obligatoire de passer par cette phase .

NEXDOC

(pas de variable)

3.23 NEXTREC

Cette instruction permet d'envoyer le record déjà constitué ,sans passer par l'instruction SEND . Ceci a pour avantage ,aussi ,d'éviter d'attendre que le lecteur coupe de lui même l'enregistrement dès qu'il a atteint la longueur défini dans la partie AXF (définition des paramètres de transmission) ; en effet si la longueur définie dans l'instruction SEND est supérieure à la longueur d'enregistrement définie dans AXF , le lecteur transmettra plusieurs enregistrements de la longueur définie dans AXF . L'instruction NEXTREC permet donc de définir la fin de chaque enregistrement ,de façon précise , évitant ainsi ,par exemple, d'avoir une zone lue ,à cheval sur deux transmissions . .

NEXTREC

Exemple : NEXTREC

Cette page a été laissée blanche intentionnellement

3.24 PRINT

Cette instruction permet d'imprimer sur le document ,en fin de lecture , une partie de l'enregistrement d'édition .Cette instruction ne peut être utilisée que sur les lecteurs équipés de l'option imprimante "interne" ; à ne pas confondre avec la sortie périphérique optionnelle de type Centronics appelée "externe" .

PRINT FROM début LENGTH longueur AT position

Début	définit la position ,dans l'enregistrement d'édition , du 1er caractère à imprimer	1 à 12000
Longueur	définit le nombre de caractère à partir de "début" à prendre dans l'enregistrement pour imprimer	1 à 120
Position	définit la distance (en mm) entre le haut du document et l'impression du 1er caractère .	1 à 290

Nota : l'impression se fait toujours à 90° par rapport aux marques horloges .

3.25 RETURN (+)

Cette instruction est utilisée pour terminer une sous-routine .Le programme retournera alors exécuter l'instruction juste après l'instruction ayant appelé cette sous-routine :
.(IF.....GOSUB ou GOSUB) .

RETURN (pas de variable)

3.26 RFIELD

Cette instruction (REPEAT FIELD) va dupliquer la définition de champ (instruction FIELD) précédente ,avec exactement les mêmes paramètres (N° de ligne,N° de colonne,Nb de données par groupe ,etc...) ,exceptés les paramètres ayant été changés par les instructions MEMORY (de 1 à 15) utilisées entre FIELD et RFIELD .

Cette instruction sera souvent utilisée pour écrire des sous-routines avec des champs relatifs .

Nota : voir les instructions MEMORY pour leurs descriptions .

RFIELD

(pas de variable)

Exemple : FIELD 1001,5,1,12,1,1,2,FE,VER,LEFT,0,0,9,0,1
 MEMORY 1 = 0
 MEMORY 4 - 4
 RFIELD

- Ce court exemple va d'abord exécuter une instruction FIELD pour décrire un champs débutant à la colonne 12 ,ligne 1 ,avec table 2 ,etc... ;et le résultat de la lecture sera écrit dans l'enregistrement d'édition en position 1001 .
- La seconde ligne met l'index d'écriture dans l' enregistrement d'édition à la prochaine adresse libre (contenue dans la mémoire 54) .Ici 1002 .
- La troisième ligne donne la colonne de début du nouveau champ par rapport au précédent (- 4) .Ici donc 12 - 4 : nouvelle colonne = 8 .
- La dernière ligne donne l'ordre de création d'un nouveau champ avec tous les paramètres définis dans l' instruction FIELD ,excepté la position dans l'enregistrement et la colonne du début du champ.

3.27 RFIELDXY

Cette instruction (REPEAT FIELDXY) va dupliquer la définition de champ (instruction FIELD) précédente ,avec exactement les mêmes paramètres (N° de ligne,N° de colonne,Nb de données par groupe ,etc...) ,exceptés les paramètres ayant été changés par les instructions MEMORY (de 1 à 15) utilisées entre FIELDXY et RFIELDXY .

Cette instruction sera souvent utilisée pour écrire des sous-routines avec des champs relatifs .

Nota : voir les instructions MEMORY pour leurs descriptions .

RFIELDXY

(pas de variable)

Exemple : FIELDXY 1001,5,1,12,1,1,2,FE,VER,LEFT,0,0,9,0,1
 MEMORY 1 = 0
 MEMORY 4 - 4
 RFIELDXY

- Ce court exemple va d'abord exécuter une instruction FIELDXY pour décrire un champs débutant à la colonne 12 ,ligne 1 ,avec table 2 ,etc... ;et le résultat de la lecture sera écrit dans l'enregistrement d'édition en position 1001 .
- La seconde ligne met l'index d'écriture dans l' enregistrement d'édition à la prochaine adresse libre (contenue dans la mémoire 54) .Ici 1002 .
- La troisième ligne donne la colonne de début du nouveau champ par rapport au précédent (- 4) .Ici donc 12 - 4 : nouvelle colonne = 8 .
- La dernière ligne donne l'ordre de création d'un nouveau champ avec tous les paramètres définis dans l' instruction FIELDXY ,excepté la position dans l'enregistrement et la colonne du début du champ.

Note : l'instruction RFIELDXY exécute aussi la recopie d'un champs défini par l'instruction FIELD ,ceci sous les même condition que la recopie d'un champs RFIELDXY (recopie avec les modifications apportées sur les mémoires 1 à 15)

3.28 SEND

Quand cette instruction est rencontrée par ADELE ,le logiciel envoie immédiatement le morceau spécifié de l'enregistrement d'édition sur la sortie série du lecteur .La première fois que l' instruction SEND est vue pour chaque document (ou après SEND 0,0) ,les caractères spécifiés comme "début de document" sont transmis .

Les caractères spécifiés comme "fin de document" seront ,quant à eux ,envoyés seulement sur une instruction SEND 0,0 ,ou une fin de programme (ex : NEXDOC)

.Cette possibilité permet d'envoyer en plusieurs fois le contenu d'un même document , faisant ainsi varier,par exemple ,les enregistrements transmis ,en fonction des résultats de la lecture ou des réponses de l'ordinateur.

SEND début fin

SEND 0 , 0

Début	définit l'adresse du 1er caractère de l'enregistrement d'édition à transmettre	0,1 à 12000
Fin	définit l'adresse du dernier caractère de l'enregistrement d'édition à transmettre	0,1 à 12000

NOTA :

- si "début" est défini à 0 ,le 1er caractère pris pour la transmission sera l'adresse 1
- si "fin" est défini à 0 ,le dernier caractère transmis sera l'adresse la plus haute dans l'enregistrement d'édition à avoir été écrit au moment du SEND
- si ensemble "début" et "fin" ont été défini à 0 ,l'enregistrement sera transmis et les caractères "fin d'enregistrement" seront immédiatement suivi des caractères "fin de document" .Une autre instruction SEND peut suivre ,pour le même document ,mais le nouvel enregistrement sera obligatoirement précédé de "début de document" , "début d'enregistrement" .

Exemple :

SEND 101,0	transmet de l'adresse 101 de l'enregistrement d'édition, jusqu'à la dernière adresse utilisée
SEND 0,15	transmet de l'adresse 1 à l'adresse 15 (incluse)
SEND 23,25	transmet 3 caractères de l'adresse 23 à 25 (incluse)
SEND 0,0	fini l'enregistrement et envoie le caractère fin de document

Note : le SEND 0,0 est souvent utilisé pour finir la lecture après un ABORT ou NOSEND

3.29 SLEEP

Cette instruction met en "sommeil" le lecteur ,qui ne prendra alors plus aucune commande ,jusqu'à la réception du mot défini dans l' instruction ACTIVE .

SLEEP

(pas de variable)

Exemple : **SLEEP**

3.30 TABLEA

Cette instruction est utilisée pour définir les tables de conversion ,des marques lues en données ,pour l'enregistrement d'édition .Ces tables sont utilisées pour convertir les marques lues en **1 caractère ASCII** .

TABLEA n , saut données,données,données,...

n	N° d'identification de la table (1 à 199)
,	Le séparateur entre n et saut doit être une virgule
saut	Le caractère donné ici indiquera que la marque doit être "sautée"
données	Le caractère ASCII représentant chaque marque est donné ici

Le paramètre "saut" est un caractère défini par l'utilisateur pour indiquer toutes les marques qui doivent être ignorées par ADELE ,ceci permet d'avoir des groupes avec les cases de marquages (alvéoles) espacées de une ou plusieurs colonnes (groupe horizontal) ou lignes (groupe vertical) .Cela signifie qu'en cas d'utilisation de l' instruction COUNTER ,si une marque est lue sur une des positions "sautées" elle ne sera ,bien évidemment ,pas comptée.

Le caractère de "saut" peut être n'importe quel caractère ASCII compris entre 20H et 7FH ,mais ne doit pas être utilisé dans les données, qui suivent .

Les données doivent être écrites sur une seule ligne et la 1ère donnée ne peut être un séparateur (CR,LF,tab,espace,virgule,point virgule) .

Le CR (carriage return ,touche "entrée") est utilisé comme caractère de fin de table.

Seuls les caractères compris entre 20H et 7FH peuvent être utilisés .

NOTA : si plus d'une marque est lue sur le même groupe ,seule la dernière est mise dans l'enregistrement d'édition .Ce phénomène peut être évité ,si on le souhaite ,lors de la description des champs en définissant "nombre maximum de marques" à 1 ,il y aura , dans ce cas, une erreur de détectée .

Exemple :

TABLEA 10,* A**B**C**D**F cette table lira des valeurs de A à F espacées toutes les 3 lignes ou colonnes .

TABLEA 23,- 123-ABC cette table attribuera 1 à la 1ère case ,2 à la 2ème ,3 à la 3ème ,la 4ème sera ignorée (ici le caractère de saut est -) ,A à la 5ème etc ...

3.31 TABLEM

Cette instruction est utilisée pour définir les messages pour les instructions IFERROR et MESSAGE .

TABLEM n , saut texte

n	N° d'identification de la table
,	Le séparateur entre n et saut doit être une virgule (2CH)
saut	Pas de fonctionnalité dans cette table (mais doit être présent)
texte	Chaîne ASCII que l'on veut obtenir .

- Le paramètre "saut" doit être n'importe quel caractère entre 20H et 7FH mais ne doit pas être utilisé dans le texte du message .

- Le texte doit être sur une seule ligne ,le 1er caractère ne peut être un CR ,LF ,tab , espace ,virgule , ou point virgule .La longueur du texte est libre et doit être terminé par un CR (touche "Entrée") .

Seuls les caractères entre 20H et 7FH peuvent être utilisés

Exemple :

TABLEM 10,* absence de marque dans le champ client

3.32 TABLEN

Cette instruction est utilisée pour définir des tables de conversion des marques lues en données ,pour l'enregistrement d'édition .Ces tables sont utilisées pour convertir les marques lues en **valeurs numériques** .

TABLEN n données,données,données

n N° d'identification de la table

Les valeurs correspondantes à une marque sont données en décimal .La fourchette est de 0 à 65500 .Les valeurs correspondantesà chaque marque lue dans un groupe sont additionnées et le résultat mis dans l'enregistrement d'édition .Le nombre de caractère écrit dans l'enregistrement dépend de la réponse à "caract/gpe" lors de la définition du champs .

Le séparateur entre deux données doit être une virgule (héxa 2C) .

La table peut ne pas être sur une seule ligne (pour une meilleure lecture par exemple) , dans ce cas la ligne doit être terminée par CR .

Pour sauter une marque dans la table ,mettre deux virgules entre données

Pour sauter deux marques ,mettre trois virgules ,etc...

La dernière ligne de la table doit finir par une donnée et un CR (touche entrée) .

Exemple :

TABLEN 4 1,,2,,4,,8,,16,,32,,64,,128,,
256,,512,,1024

Cette table lira des valeurs binaires toutes les deux colonnes ou lignes .

3.331 TABLES

Cette instruction est utilisée pour définir des tables de conversion des marques lues en données ,pour l'enregistrement d'édition .Ces tables sont utilisées pour convertir les marques lues en **chaîne de caractères**

```
TABLES      n , long , saut , fin CR
données CR
données CR
....
données CR
fin        CR
```

n	N° d'identification de la table
long	Nombre de caractères à attribuer à chaque marque dans l'enregistrement
saut	Caractère pour indiquer un saut entre deux marques à lire
fin	Caractère pour marquer la fin de la définition de la table

- La valeur correspondante à chaque marque est donnée en ASCII .Pour une marque , il est possible d'affecter plusieurs caractères ASCII .Il est possible ,par exemple , d'attribuer un mot ou une phrase pour n'importe quelle marque d'un champ .

IMPORTANT :

- un item est la donnée attribuée à chaque marque
- à l'intérieur d'une table chaque item donnera la même longueur dans l'enregistrement
- chaque ligne correspond à un item .Un CR **doit** finir chaque ligne
- pour chaque ligne le 1er caractère ne peut être un séparateur
- les points virgules ne sont pas autorisés ,ils définissent un début de commentaire
- les virgules sont autorisées
- si les données sont plus longues que "long" ,elles seront coupées par ADELE
- si les données sont plus courtes que "long" ,elles seront complétées par des espaces
- si le premier caractère d'une ligne est un "saut" ,la marque sera ignorée
- la fin de table doit être une ligne commençant par le caractère "fin"
- si plus d'une marque est lue dans le groupe ,seule la donnée correspondante à la dernière sera mise dans l'enregistrement d'édition .

Exemple :

```
TABLES      2 , 5 , * , / ; table de couleur  
  
            Vert                ;couleur de la 1ère marque  
            Rouge  
            *  
            Jaune  
            Bleu  
            /
```

Dans cet exemple ,si la 1ère marque est lue ,la donnée écrite dans l'enregistrement d'édition sera "VERT " ,l'espace à la fin servira à cadrer sur 5 caractères définis dans la table ;la 2ème marque générera "Rouge" ,la troisième position dans le champs est à sauter ;la 4ème "Jaune" ,etc..

3.34 TABLET

Cette instruction est utilisée pour définir une table de translation .Cette table sert à convertir une ou plusieurs marques en un caractère ASCII .Cela permet d'encoder jusqu'à huit marques en un seul caractère .

La translation s'effectue en trois étapes :

1°) La valeur des marques du groupe sont additionnées dans une table numérique ;en utilisant les valeur déclarées dans la première ligne de la table (en général 1,2,4,8,16,32,64,...)

2°) ADELE recherche si la valeur trouvée en 1°) existe dans la seconde partie de la table ,si elle n'existe pas la valeur par défaut définie dans "def" sera écrite dans l'enregistrement avec en plus le status "nombre de marque" mis en erreur .Il suffira ensuite de contrôler et de donner l'action à exécuter ,suite à ce status ,dans l' instruction IFERROR utilisée juste avant l'appel de cette table .

3°) Le caractère trouvé est écrit dans l'enregistrement d'édition .

TABLET n données,données,données

DEF 'd'

n1 'a1'

n2 'a2'

... ...

nx 'ax'

/

n N° d'identification de la table

données De 2 à 8 valeurs numériques ,la somme des données ne peut dépasser 255

d Caractère ASCII à écrire si aucune combinaison n n'est trouvée

n1 à nx Somme à convertir

a1 à ax Caractère à écrire pour chaque somme en regard

Le séparateur entre chaque "données" doit être une virgule (2CH)

Pour sauter une marque dans la table ,mettre deux virgules entre données

Pour sauter deux marques ,mettre trois virgules ,etc...

La dernière ligne doit être un / .

Exemple :

```
TABLET 4 1,,2,,4,,8,,16
DEF  '?'
3      '0'
5      '1'
6      '2'
9      '3'
10     '4'
12     '5'
17     '6'
18     '7'
20     '8'
24     '9'
/
```

Cette table est une définition dite 2 parmi 5 .

La 1ère ligne indique que cette table (N° 4) ,comporte 5 alvéoles ayant les valeurs respectives 1 à 16 .Ces alvéoles sont réparties 1 colonne ou 1 ligne sur 2 (présence de deux virgules entre chaque valeur) .

Le caractère mis dans l'enregistrement ,au cas ou aucun des cas définis à partir de la 3ème ligne n'est trouvé sera un "?" .

Si les alvéoles 1 et 2 sont cochées alors la somme donnera 3 (1+2) ,la valeur mise dans l'enregistrement sera 0 .

Si les alvéoles 1 et 3 sont cochées alors la somme donnera 5 (1+4) ,la valeur mise dans l'enregistrement sera 1 .

Si les alvéoles 4 et 5 sont cochées alors la somme donnera 24 (16+8) ,la valeur mise dans l'enregistrement sera 9 .

3.35 WAITKEY

Cette instruction bloque le déroulement du programme jusqu'à la pression du bouton de la face avant du lecteur ,par l'opérateur .L'exécution des instructions suivantes du programme reprend dès cette pression .

WAITKEY

(pas de variable)

3.36 WRITEC

Cette instruction permet d'écrire une chaîne jusqu'à 20 caractères , dans l'enregistrement d'édition .La chaîne doit commencer par ' ou " et finir par le même caractère .

Le caractère défini comme délimiteur ne doit pas se retrouver dans la chaîne .

WRITEC dest "chaîne" CR ou **WRITEC dest 'chaîne' CR**

dest	adresse dans l'enregistrement où écrire la chaîne (0,1 à 12000)
chaîne	chaîne de caractères à écrire dans l'enregistrement

- Si "dest" est égal à 0 ,la chaîne sera écrite à l'adresse contenue dans MEMORY 54 (mémoire de l'index d'écriture) ,juste après l'adresse la plus haute déjà écrite dans l'enregistrement d'édition .

Exemple :

WRITEC	100	'Erreur dans "zone client"'	bonne formulation
WRITEC	100	"Reçu caractère 'A'"	bonne formulation
WRITEC	100	"Erreur de : "cadrage""	mauvaise formulation

APPENDICE A : Liste des erreurs

Les erreurs systèmes suivantes peuvent être envoyées par les lecteurs .Ces erreurs indiquent aussi bien des états du lecteur (magasin vide) ,que des problèmes mécaniques (bourrage sous la tête) .

Ces messages sont précédés et suivis par les caractères définis ,dans les paramètres de communication (voir chapitre 2) ,comme caractères de début et de fin de "error message" .

1°) Messages "matériel" (ex : entraînement ,cellules ,...)

Erreur N°	MESSAGES (écran 16 caractères)	MESSAGES (écran 4 caract.)	
1	PROB BAC D'APPEL	EMAT(3 essais)	feuilles non entraînées
2	PROB AV TETE	JAMB	bouillage avant la tête
3	PROB SOUS TETE	JAMH	bouillage sous la tête
4	PROB APRES TETE	JAMA	bouillage après la tête
5	PROB EN SORTIE	SORT	bouillage en éjection
6	DOCUMENT SVP	PAPR	magasin d'entrée vide
7	BAC REJET PLEIN		bac de rejets plein
8	BAC BON PLEIN		bac des bons plein
9	TROP COURT		document trop court
10	TROP FIN		document trop mince
11	TROP EPAIS		document trop épais
12	TROP LONG		document trop long
13	PAPIER		document
14	INCORRECT PARITY		parité incorrecte
15	BAUDRATE ?		vitesse de transmission ?
16	SENSIBILITE		sensibilité
17	LOW BATTERY		pile en fin de vie
18	DONNEES PERDUES		paramétrages perdus
19	RIEN A EJECTER		pas de doc. à transmettre
20	BOUCHON		transport non libre
21	PAS DE REF NOIRE	BLAC	référence noire mauvaise
22	PAS DE DECODEUR		pas de décodeur code barre
23	PLATEAU HS		ascenseur d'entrée hors servi
24	CALIBRATE		calibration
64	PAUSE	PAUS	pause
128	PAUSE		pause

LISTE DES ERREURS *manuel de programmation*

2°) Messages ADELE

Erreur N°	MESSAGES (écran 16 caractères)	MESSAGES (écran 4 caract.)	
	TOUCHE SVP	TCHE	appuyer sur touche face av.
	AUR	INIT	initialisation
	COMMANDE SVP	CMDE	attente de commande
	EN LECTURE	LECT	lecture
	EN TRAITEMENT	TRAV	traitement en cours
		XMIT	transmission en cours
	SELECTION SVP	SELE	attente de selection
		CONT	continuer
	DOCUMENT SVP	DOCU	mettre un document
	PRET A LIRE	OK	prêt à lire
	PARAMETRE SVP	PARA	attent un téléchargement
	EN ORDRE RECU	OK	fin de téléchargement
	ERREUR LOGIQUE	LOGI	erreur de lecture (logique)
	ACTIVE	ACTV	attend mot de passe
	NIVEAU LECT =	S =	sensibilité = (de 1 à 7)
	VERIF CELLULE		cellule obstruée (1 à 5)
	XXXX TH LINE	XXXX	ligne en cours de télécharg.

APPENDICE B : Table ASCII des caractères

Octal	Décimal	Héxa	Caract.	Ctrl + to	description
000	0	0	NUL	@	nul
001	1	1	SOH	A	début de synchro
002	2	2	STX	B	début de texte
003	3	3	ETX	C	fin de texte
004	4	4	EOT	D	fin de transmission
005	5	5	ENQ	E	enquiry ,aussi WRU
006	6	6	ACK	F	acquiescement (acknowledge)
007	7	7	BEL	G	sonnerie (ring bell)
010	8	8	BS	H	backspace
011	9	9	HT	I	tabulation horizontale
012	10	A	LF	J	line feed (ligne suivante)
013	11	B	VT	K	tabulation verticale
014	12	C	FF	L	form feed (page suivante)
015	13	D	CR	M	carriage return (touche entrée)
016	14	E	SO	N	shift out
017	15	F	SI	O	shift in
020	16	10	DLE	P	data line escape
021	17	11	DC1	Q	XON device ctrl 1
022	18	12	DC2	R	device ctrl 2
023	19	13	DC3	S	device ctrl 3 XOFF
024	20	14	DC4	T	device ctrl 4
025	21	15	NAK	U	non acquiescement
026	22	16	SYN	V	synchronous idle
027	23	17	ETB	W	fin de transmission de bloc
030	24	18	CAN	X	cancel
031	25	19	EM	Y	fin de medium
032	26	1A	SUB	Z	substitute
033	27	1B	ESC	[escape (échappement)
034	28	1C	FS	\	file separator
035	29	1D	GS]	group separator
036	30	1E	RS	^	record separator
037	31	1F	US	-	unit separator
040	32	20	SP		espace
041	33	21	!		
042	34	22	"		
043	35	23	#		
044	36	24	\$		
045	37	25	%		
046	38	26	&		
047	39	27	'		apostrophe
050	40	28	(
051	41	29)		
052	42	2A	*		
053	43	2B	+		

TABLE ASCII*manuel de programmation*

Octal	Décimal	Héxa	Caract.	Ctrl to	description
054	44	2C	,		virgule
055	45	2D	-		signe moins
056	46	2E	.		point
057	47	2F	/		
060	48	30	0		
061	49	31	1		
062	50	32	2		
063	51	33	3		
064	52	34	4		
065	53	35	5		
066	54	36	6		
067	55	37	7		
070	56	38	8		
071	57	39	9		
072	58	3A	:		deux points
073	59	3B	;		point virgule
074	60	3C	<		
075	61	3D	=		
076	62	3E	>		
077	63	3F	?		
100	64	40	@		
101	65	41	A		
102	66	42	B		
103	67	43	C		
104	68	44	D		
105	69	45	E		
106	70	46	F		
107	71	47	G		
110	72	48	H		
111	73	49	I		
112	74	4A	J		
113	75	4B	K		
114	76	4C	L		
115	77	4D	M		
116	78	4E	N		
117	79	4F	O		
120	80	50	P		
121	81	51	Q		
122	82	52	R		
123	83	53	S		
124	84	54	T		
125	85	55	U		
126	86	56	V		
127	87	57	W		
130	88	58	X		

Octal	décimal	Héxa	Carat.	Ctrl to	Description
131	89	59	Y		
132	90	5A	E		
133	91	5B	[
134	92	5C	\		anti slash
135	93	5D]		
136	94	5E	^		
137	95	5F	_		souligné
140	96	60	`		accent grave
141	97	61	a		
142	98	62	b		
143	99	63	c		
144	100	64	d		
145	101	65	e		
146	102	66	f		
147	103	67	g		
150	104	68	h		
151	105	69	i		
152	106	6A	j		
153	107	6B	k		
154	108	6C	l		
155	109	6D	m		
156	110	6E	n		
157	111	6F	o		
160	112	70	p		
161	113	71	q		
162	114	72	r		
163	115	73	s		
164	116	74	t		
165	117	75	u		
166	118	76	v		
167	119	77	w		
170	120	78	x		
171	121	79	y		
172	122	7A	z		
173	123	7B	{		
174	124	7C	 		
175	125	7D	}		
176	126	7E	~		
177	127	7F	DEL		delete

APPENDICE C : Explication des termes utilisés dans ce manuel

Alvéole	: petite case prédéfinie sur le document ,matérialisant l'endroit à remplir .																
Marque horloge	: marque noire préimprimée à droite du document .Elles donnent au lecteur les ordres de lecture (début et fin) .Chaque marque horloge représente une ligne de lecture .																
Colonne	: suivant le type de tête de lecture du lecteur ,on obtient 12,32,40 ou 48 colonnes de lecture ,la distance entre chaque colonne dépend aussi de ce type .En standard 40 colonnes espacées de 1/5 de pouce																
Champs	: ensemble de un ou plusieurs groupes ,chaque groupe doit avoir la même longueur et le même contenu (type de données).Un champ est toujours rectangulaire .																
Groupe	: une ou plusieurs alvéoles ,sur la même ligne ou la même colonne , contenant des données à lire .																
Ligne	: une ligne est définie par une marque horloge ;la distance entre deux lignes est libre .Chaque marque horloge (y compris celles sans données à lire en regard) ,doit être comptée pour connaître le N° de ligne .																
Marque	: marque en encre lisible ,à l'intérieur d'une alvéole .																
Séparateurs	: les caractères suivants sont utilisés comme séparateur : <table><tr><td>Carriage return</td><td>0Dh</td></tr><tr><td>Line feed</td><td>0Ah</td></tr><tr><td>Espace</td><td>20h</td></tr><tr><td>Tabulation</td><td>09h</td></tr><tr><td>Virgule</td><td>2Ch</td></tr><tr><td>Point virgule</td><td>3Bh</td></tr><tr><td>Parenthèse gauche</td><td>28h</td></tr><tr><td>Parenthèse droite</td><td>29h</td></tr></table>	Carriage return	0Dh	Line feed	0Ah	Espace	20h	Tabulation	09h	Virgule	2Ch	Point virgule	3Bh	Parenthèse gauche	28h	Parenthèse droite	29h
Carriage return	0Dh																
Line feed	0Ah																
Espace	20h																
Tabulation	09h																
Virgule	2Ch																
Point virgule	3Bh																
Parenthèse gauche	28h																
Parenthèse droite	29h																

1°) Les méthodes :

Il est possible de paramétrer son lecteur sous ADELE ,par deux moyens ;soit par l'utilisation du logiciel AXMPC ,soit directement par l'éditeur habituellement utilisé sur le site client . En effet les lignes de programmation ,sont en ASCII sans aucun caractère de tabulation pouvant perturber certains éditeur .Le moyen le plus facile restant toutefois l'utilisation d'AXMPC ,qui de par la convivialité de ses écrans et surtout de ses contrôles syntaxistes guide et surveille le déroulement de la programmation .

Note : du fait de la possibilité de travailler le fichier ADELE sous éditeur ,cela permet pour les utilisateurs dont le but final est la connection du lecteur sur un autre système que DOS (ex : UNIX) de développer sous DOS, mais une fois le fichier transmis sur le système hôte , de pouvoir finaliser, ou modifier, l'application sans avoir à revenir sous DOS .

2°) La programmation :

Celle-ci se divise en trois parties (la communication ,le programme ,les tables) ;

a) habituellement la 1ère partie n'a pas à être modifiée ,hormis la longueur de l'enregistrement, le niveau sonore du bip et le caractère de remplissage .**Toute autre modification est même formellement interdite** pour l'utilisateur **souhaitant utiliser le programme de lecture intégré dans AXMPC** ,celui-ci ne travaillant qu'avec les paramètres définis par défaut .Les autres paramètres ne pourront donc être touchés que pour les programmeurs souhaitant développer un programme de lecture en dehors d'AXMPC (ex : monde UNIX) .

b) la partie programme :

- déroulement du paramétrage :

FILL : ne pas oublier cette instruction en 1ère ligne , sinon le lecteur garde en mémoire l'enregistrement de la dernière feuille lue .Cette instruction doit se faire sur l'intégralité de l'enregistrement ,ou alors seulement sur une partie dans la cas ou il est souhaité de garder une partie de l'enregistrement précédent .

WRITEC : toujours initialiser toutes les variables, que l'on aura besoin ensuite ,en début de programme .On se réserve généralement les adresses, dans l'enregistrement, de 1 à 100 .Ceci évite ainsi ,plus tard d'écraser ces variables, par des écritures à la même adresse .Le fait de les écrire toutes dans la même portion de l'enregistrement ,permet aussi une visualisation rapide des adresses libres , pour insérer de nouveau WRITEC.

IFERROR : **avant toute lecture (y compris zone code barre) ,il est impératif d'avoir un IFERROR ,permettant d'initialiser les tables d'erreur ; même si la 1ère zone ne doit pas générer d'erreur .**

FIELD : le premier FIELD devant détecter le type de feuille (en cas de multifeuille); le type de feuille étant donné par la zone préimprimée en noire (généralement en haut à droite du document) .

GOTO : dès la détection du type de feuille il conviendra d'orienter son programme vers un LABEL où l'on retrouvera les instructions convenant au type de feuille détectée .

LABEL : ces adresses de saut (pour les GOTO et GOSUB) sont limités au nombre de 199 .Même si pour des questions d'esthétisme du programme ,ces labels seront écrit par ordre de numéro ,ceci n'est pas une obligation pour le lecteur généralement le LABEL 199 contient la fin du programme c'est à dire les instructions SEND et NEXDOC .

CLOCK : il est impératif de contrôler le nombre de marque horloge lues par le lecteur avant toute suite de la décodification ;en effet toute mauvaises lectures de ces marques entrainerait un décalage des zones de lectures .

IFMEMORY :le contrôle des marques horloges peut se faire aussi en comparant la mémoire 55, pour le recto ,au nombre attendu de marques (méthode plus rapide que la procédure CLOCK) ; ou à la mémoire 56 pour le verso .

Ensuite on retrouvera une suite d'instructions en fonction de la lecture et des contrôles souhaités .

MESSAGE : une attention particulière doit être apportée à l'option ABORT dans ces messages ,il est obligatoire de passer ensuite par une instruction NEXDOC et SEND (voir explication des instructions) pour permettre de passer la main correctement à la lecture de la feuille suivante .Dans le cas où l'on a sa fin habituelle de lecture en LABEL 199 il suffit après la ligne MESSAGE de faire un GOTO sur ce label ; ne pas s'inquiéter dans ce cas l'instruction send servira de "clotûre" mais aucune transmission (hormis le message) ne sera faite .

Lors de l'utilisation de cette instruction en vue d'un INPUT,avec le logiciel AXMPC il convient de faire précéder le message envoyé à l'ordinateur (pour demander l'entrée clavier) ,d'un \$ (ex : \$Date du jour : ?) .Dans la même utilisation ,l'action conseillée est "SEND TEXT" ,car c'est la seule qui ne monte pas le drapeau d'erreur (d'ou pas de risque de confusion pour le stockage bon/mauvais

le

SEND : en fin de programme cette ligne permettra de transmettre la portion d'enregistrement souhaitée (sur les 12000 caractères disponibles) ainsi que de fermer la transmission (envoi du caractère de fin d'enregistrement) .

NEXDOC : dernière ligne d'instruction demande la lecture du document suivant .

Note : il est important de conserver à l'esprit que **chaque manipulation** sur l'enregistrement fait **évoluer le pointeur** (conservé dans la MEMORY 54) de la prochaine adresse libre dans l'enregistrement .Attention donc dans le cas de travail avec "Adresse flottante" (réponse 0 pour adresse dans l'enregistrement) .C'est pour cette raison que généralement on se réserve les positions 1 à 500 de l'enregistrement pour les zones dites de travail (WRITEC ,stockage ponctuel de données ,etc ...) , évitant ainsi les interférences avec la fabrication de l'enregistrement final, devant être transmis .

c) la partie des tables

Les tables divisées en 5 catégories (ASCII,NUMERIQUE, STRING, TRANSCODAGE, et MESSAGE) sont limitées à 200 ,par contre le nombre de données contenues dans celles-ci n'est pas limité (hormis l'éventuelle saturation de la mémoire du lecteur) .Il est donc conseillé de ne pas utiliser une table pour une valeur ,mais plutôt d'introduire cette valeur dans une autre table et de jouer dans l'instruction FIELD sur les options longueur du groupe et 1ère position dans la table

Généralement pour des raisons de clarté de lecture du programme ,on réserve les tables 1 à 100 pour les tables de données et de 101 à 199 pour les tables de messages .

3°) Conseils :

Il est fréquent que pour toutes les applications d'un même utilisateur ,beaucoup de ligne de programmation se retrouvent ;une astuce pour éviter de réécrire ces lignes consiste à "préformater" son programme dans le fichier NOUVEAU.AUR .Ce fichier sera alors celui ouvert au début de chaque nouvelle application ; on aura ainsi pris soin de lui introduire par exemple les sous routines , les tables ,etc... utilisées à chaque application , évitant ainsi tout nouveau développement , de ces lignes .

4°) Utilisation d'un lecteur ADELE avec AXMPC :

Les fonctionnalités d'un lecteur ADELE (contrôles ,sous routines ,etc...) ,permettent avec l'utilisation d'AXMPC (correction en ligne ,entrées clavier ,gestion des fichiers ,etc...), dans la plupart des cas d'éviter de réécrire un programme applicatif .L'utilisation de certaines options AXMPC necessitent toutefois certaines attentions :

a) Correction en ligne : il faut faire précéder les données du caractère de "status" (paramétrage dans la partie AXF : "envoi du caractère de status") .Le curseur de la correction ne s'arrête que sur les ? ,il faut donc le prévoir dans les IFERROR .La correction est demandée que si la feuille a été déclarée mauvaise ,y penser dans IFERROR , MESSAGE ,CHECK .

b) Entrée clavier : le logiciel AXMPC **n'ouvre la fenêtre** de saisie clavier que sur la réception d'un MESSAGE commençant par \$.Il faut donc faire précéder l'instruction INPUT d'un passage par un MESSAGE

c) Lancement d'AXMPC en automatique ,par exemple en venant d'un menu "client" , évitant ainsi les manipulations ,chargement du programme de lecture ,lancement des options, lancement de la lecture .La syntaxe doit se présenter sous la forme :

AXMPC [/C:n] [/P:xxx] [/D:xxx] [/E:xxx] [/R:xxx] [L+] [/O+]

C: 1,2 pour com1 ou com2 (ou adresse en héxa ex: /C:03E8)

P: nom du fichier paramètre , avec ou sans path

D: Fichier pour écriture des documents "bons" ,avec ou sans path

E: Fichier pour écriture des documents "erronnés", avec ou sans path

R: Fichier des messages d'erreur , avec ou sans path

L+ : Demande de téléchargement

O+ : Lancement automatique de la lecture

NOTE : toutes les écritures dans cette commande doivent être en majuscule .

APPENDICE E : RESUME PAR FONCTIONNALITE

Mots de passe et identification

ACTIVE	donne le nom du mot de passe et valide le dialogue
SLEEP	verouillage du dialogue lecteur / ordinateur
NAME	nom que portera le programme et que le lecteur retransmettra sur demande (permet d'identifier le programme en mémoire)
COUNTER	donne un numéro d'identification à un début de zone (chaque groupe de la zone lue portera un numéro d'identification incrémenté à partir de celui-ci)

Instructions de lecture

<u>NOM de l'INSTRUCTION</u>	<u>ORIGINE des DONNEES</u>	<u>AFFECTATION</u>
BARCODE	Tête de lecture code barre	Adresse dans l'enregistrement
CLOCK	Colonne marques horloges	Adresse dans l'enregistrement
FIELD	Tête de lecture marques	Adresse dans l'enregistrement
RFIELD	Tête de lecture marques	Adresse dans l'enregistrement
FIELDXY	Tête de lecture marques	Adresse dans l'enregistrement
RFIELDXY	Tête de lecture marques	Adresse dans l'enregistrement

Manipulations et écritures vers l'enregistrement

<u>NOM de l'INSTRUCTION</u>	<u>ORIGINE des DONNEES</u>	<u>AFFECTATION</u>
COPY	Adresse dans l'enregistrement	Adresse dans l'enregistrement
FILL	Constante ASCII	Adresse dans l'enregistrement
INPUT	Ligne V24	Adresse dans l'enregistrement
MEMORY	Contenu de la mémoire	Adresse dans l'enregistrement
WRITEC	Texte ASCII	Adresse dans l'enregistrement

Manipulations et écritures à partir de l'enregistrement

<u>NOM de l'INSTRUCTION</u>	<u>ORIGINE des DONNEES</u>	<u>AFFECTATION</u>
DISBUF	Section de l'enregistrement	Ecran du lecteur
IF ERROR	Section de l'enregistrement	Ligne V24 ou imprimante exter.
IF GOTO /GOSUB	Comparaison de 2 sections	Label du programme
MEMORY	Section de l'enregistrement	Contenu de la mémoire
MESSAGE	Section de l'enregistrement	Ligne V24
NEXREC	Enregistrement déjà écrit	Ligne V24
PRINT	Section de l'enregistrement	Imprimante interne
SEND	Section de l'enregistrement	Ligne V24

APPENDICE E : RESUME PAR FONCTIONNALITE

Actions sur le lecteur pour l'opérateur

<u>NOM de l'INSTRUCTION</u>	<u>ORIGINE des DONNEES</u>	<u>AFFECTATION</u>
DISPLAY	Texte ASCII	Ecran du lecteur
EJECT		Ejection physique du document pour permettre une correction clavier écran
WAITKEY		Demande de pression sur touche face avant .

Les instructions de sauts conditionnels ou non

<u>NOM de l'INSTRUCTION</u>	<u>CONDITIONS</u>	<u>DESTINATION</u>
GOTO	Immédiat	N° de LABEL
GOSUB	Immédiat	N° de LABEL
IF GOTO	Comparaison de 2 sections d'enregistrement (< > = #)	N° de LABEL
IF GOSUB	Comparaison de 2 sections d'enregistrement (< > = #)	N° de LABEL
IF MEMORY	Comparaison d'1 mémoire à une autre ou à une constante (< > = #)	N° de LABEL
LABEL		Adresse pour les sauts
RETURN	Fin d'un GOSUB	Ligne suivante du GOSUB

Les contrôles et rejets

<u>NOM de l'INSTRUCTION</u>	<u>CONDITIONS</u>	<u>DESTINATION</u>
IF ERROR	Rejets sur une zone lue (obligatoire ,vide ,....)	Table de message vers V24 ou imprimante externe
IF GOTO	Comparaison de 2 sections d'enregistrement (< > = #)	N° de LABEL
IF GOSUB	Comparaison de 2 sections d'enregistrement (< > = #)	N° de LABEL
IF MEMORY	Comparaison d'1 mémoire à une autre ou à une constante (< > = #)	N° de LABEL

APPENDICE E : RESUME PAR FONCTIONNALITE

Les calculs

<u>NOM de l'INSTRUCTION</u>	<u>ORIGINE DONNEES</u>	<u>DESTINATION</u>
MEMORY XXX	Une mémoire , une constante ou une section de l'enregistrement (+, -, = ,from)	La mémoire XXX
IF MEMORY		N° de LABEL

Les tables de données

<u>NOM de la TABLE</u>	<u>TYPE DE DONNEES</u>	<u>APPELEE par ...</u>
TABLE A (ASCII)	1 alvéole = 1 caractère	Instruction FIELD
TABLE N (Numérique)	Toute les marques du groupe doivent être additionnées	Instruction FIELD
TABLE M (Message)	Un message	Instructions IF ERROR , MESSAGE,
TABLE S (String)	1 alvéole = plusieurs caractères	Instruction FIELD
TABLE T (Transcription)	Chaque combinaison de marques dans un groupe donne un caractère particulier	

APPENDICE E : RESUME PAR INSTRUCTION

INSTRUCTION	ORIGINE des DONNEES	DESTINATION	FONCTION
ACTIVE	Ligne de programme (nom du mot de passe)		Donne et active le mot de passe
BARCODE	Tête de lecture code barre	Adresse dans l'enregistrement	Lecture d'une zone barcode
CHECK	Portion de l'enregistrement		Permet de faire un contrôle de modulo sur une portion (numérique) de l'enregistrement
CLOCK	Cellule marques horloges	Adresse dans l'enregistrement	Envoi du nombre de marques horloges lues
COPY	Portion d'enregistrement	Portion d'enregistrement	Copie un morceau d'enregistrement vers un autre
COUNTER	Ligne de programme (N° initial)		Initialise le N° d'ordre du 1er groupe d'une zone
DISPLAY	Ligne de programme	Ecran du lecteur	Donne le contenu du message à afficher
DISBUF	Une portion de l'enregistrement	Ecran du lecteur	Envoi une portion de l'enregistrement vers l'écran du lecteur
EJECT			Ejection du document pour permettre une correction clavier écran
FIELD	Tête de lecture des marques	Adresse dans l'enregistrement	Défini la position d'une zone à lire , son contenu et ses contrôles
FIELDXY	Tête de lecture des marques	Adresse dans l'enregistrement	Défini la position d'une zone à lire dont le groupe de base n'est pas d'un seul alignement horizontal ou vertical , ainsi que son contenu et ses controles
FILL	Ligne de programme (1 caractère ASCII)	Portion d'enregistrement	Copie sur une portion d'enregistrement une suite d'1 caractère
GOTO		N° d'un LABEL	Exécute un saut à un LABEL défini (N°)

APPENDICE E : RESUME PAR INSTRUCTION

INSTRUCTION	ORIGINE des DONNEES	DESTINATION	FONCTION
GOSUB		N° de LABEL	Exécute un saut à un LABEL défini (N°) , avec retour (en fin de sous routine)
IF GOTO	2 portions d'enregistrement	N° de LABEL	Compare 2 portions d'enregistrement et si condition "vraie" saute au N° de LABEL défini
IF GOSUB	2 portions d'enregistrement	N° de LABEL	Idem IF GOTO mais avec retour en fin de sous routine
IFERROR	Une table de message ou une portion de l'enregistrement	Ligne V24 ou imprimante externe , ainsi que case de selection sur lecteur	Donne les contrôles à effectuer sur la (les) zone(s) lue(s) ,la case d'ejection ,et le message d'erreur à transmettre
IFMEMORY	Le contenu d'une mémoire	N° de LABEL	Compare une mémoire à une autre ou à une constante et si condition "vraie" saute (GOTO, GOSUB) à 1 LABEL
INPUT	Ligne V24	Adresse dans l'enregistrement	Suspension du programme et attente de données venant de l'ordinateur
LABEL			Donne un N° d'adresse pour les lignes programmes qui suivent
MEMORY	Une constante ,une autre mémoire ou une portion de l'enregistrement	Le contenu de cette mémoire ou une section de l'enregistrement	1°)Permet le calcul sur une mémoire (+,-,=) 2°)Permet de modifier une mémoire des définitions de la zone venant d'être lue ,en vue de faire une recopie de zone 3°)Permet de récupérer le contenu des mémoires techniques

APPENDICE E : RESUME PAR INSTRUCTION

INSTRUCTION	ORIGINE des DONNEES	DESTINATION	FONCTION
MESSAGE	Table message ou portion de l'enregistrement	Ligne V24 ou imprimante externe	Envoie d'un message vers l'ordinateur ou l'imprimante connectée sur le lecteur
NAME	Nom du programme	Ligne V24 et écran du lecteur	Envoi le nom du programme pour vérifier ce qui est déjà téléchargé
NEXTDOC			Indique la fin de la lecture et le passage au document suivant
NEXTREC	Portion de l'enregistrement	Ligne V24	Envoi immédiat des données lues sans attendre l'instruction SEND
PAUSE	Ligne de programme		Suspension du programme pendant un temps défini
PRINT	Une portion de l'enregistrement	Imprimante interne	Impression d'un morceau de l'enregistrement sur le document qui vient d'être lu
RETURN			Fin d'une instruction GOSUB et retour à la ligne programme après celui-ci
RFIELD	Tête de lecture de marques	Adresse dans l'enregistrement	Recopie la zone FIELD précédente (à partir des modifications faites dans les mémoires 1 à 15)
RFIELDXY	Tête de lecture de marques	Adresse dans l'enregistrement	Recopie la zone RFIELDXY précédente (à partir des modifications faites dans les mémoires 1 à 15)
SEND	Portion de l'enregistrement	Ligne V24	Envoi vers l'ordinateur de la portion d'enregistrement sélectionné
SLEEP			Blocage du lecteur jusqu'à la réception du mot de passe

APPENDICE E : RESUME PAR INSTRUCTION

INSTRUCTION	ORIGINE des DONNEES	DESTINATION	FONCTION
TABLEA		Affectation vers les instructions FIELD	Table ASCII (1 alvéole = 1 caractère) . Définie les valeurs de chaque alvéole des zones lues
TABLEM		Affectation vers les instructions MESSAGE et IFERROR	Table message ,donne le libellé des messages envoyés vers l'ordinateur ou l'imprimante
TABLEN		Affectation vers les instructions FIELD	Table numérique (addition de toutes les marques cochées dans un même groupe)
TABLES		Affectation vers les instructions FIELD	Table string (1 alvéole = plusieurs caractères dans l'enregistrement)
TABLET		Affectation vers les instructions FIELD	Table transcodage (plusieurs alvéoles = 1 seul caractère dans l'enregistrement)
WAITKEY		Touche du lecteur	Suspend le programme jusqu'à la pression ,par l'opérateur de la touche lecteur
WRITEC	Ligne de programme	Adresse dans l'enregistrement	Ecrit un texte fixe à une adresse dans l'enregistrement (permet de faire des insertions , des comparaisons)

APPENDICE F : EXEMPLE DE PROGRAMMATION

@AXF

caractères de début de document	01 FF FF FF
caractères de début d'enregistrement	02 FF FF FF
caractères de fin d'enregistrement	0D 0A 03 FF
caractères de fin de document	04 FF FF FF
caractères de début de message d'erreur	15 FF FF FF
caractères de fin de message d'erreur	0D 0A 03 FF
caractère de remplissage	2E
quantité de caractères de synchro	00
caractère de synchro	2D
longueur d'enregistrement lsb	3C
longueur d'enregistrement msb	00
vitesse de transmission baudrate	05
longueur data sept ou huit bits	03
validation parité	00
parité paire ou impaire	01
nombre de stop bits un ou deux	01
envoi du nb marques lues oui non	00
envoi du nb d'horloges lues oui non	00
son case bonne	00
son case rejet	00
envoi prompt à la mise secteur oui non	00
envoi des messages d'erreur oui non	01
attente d'un écho oui non	00
délai avant envoi données (cinquante ms)	50
délai après chaque caractère (une ms)	01
délai après chaque record (une ms)	01
caractère pour réémission enregistrement	52
numéro de programme	00
vérification marques horloges	00
caractère pour appeler un document	4C
caractère pour appeler enregist. suivant	00
caractère pour envoi case bonne	00
caractère pour envoi case rejet	00
distance avant arrêt papier	00
caractère pour retélécharger	1B
caractère pour demande de version	56
caractère demande d'impression interne	50
saut du caractère de status	00
longueur de l'enregistrement d'édition	DC ; = 1500 limitation des 12000
reserve	05

EXEMPLE DE PROGRAMMATION *manuel de programmation*

APPENDICE F : EXEMPLE DE PROGRAMMATION

Note : ce programme n'a pas le but d'être LE modèle parfait ,mais plutôt d'utiliser de nombreuses instructions ,pour les mettre en situation réelle ; permettant ainsi d'avoir un maximum d'exemples applicatifs .Ainsi ,par exemple , le même type de zones à lire seront, parfois , traitées différemment .

@PROGRAM

```
;                                DEBUT DE PROGRAMME
NAME      ' COURS '
;
;                                remplissage de l'enregistrement d'édition avec des points
;
FILL      FROM 0 TO 1500 WITH 2E
;
=====
;                                écriture dans la zone technique
;
;                                I) nombre de marques horloges feuille 1
;
WRITEC    3 '030' ; le contrôle des horloges se fait toujours sur 3 chiffres avec
;l'instruction CLOCK
;
;                                II) numéro de feuille
;
WRITEC    9 '1'
WRITEC    10 '2'
;
;                                III) Pour le contrôle de non lecture du code a barre
;
WRITEC    11 '?????????'
;
;                                IV) Pour le contrôle produit
;                                adresse 70 et 71 protégées
WRITEC    80 '010204081631'
WRITEC    200 't.bon bon moyen insuf non ob'
;
=====;
;                                marques de cadrage lues , obligatoires mais non envoyées
;
IFERROR   100 SERIAL 3F 0 0 0 0 0 0 ABORT ; si non cochées ABORT : pas d'envoi
;                                ; de données
FIELD    1 1 1 38 1 1 1 20 HOR LEFT 0 1 1 1 1 1
FIELD    2 1 30 38 1 1 1 20 HOR LEFT 0 1 1 1 1 1
```

APPENDICE F : EXEMPLE DE PROGRAMMATION

```

=====
;
;      début de décodification de la feuille
;
=====
;
;      zone identifiatrice
;
;
IFERROR 102 SERIAL 3F 0 0 0 0 0 0 ABORT
;
;
FIELD 1000 1 1 4 1 1 2 20 HOR LEFT 0 1 1 0 1
IF      ()1000 = ()9 LENGTH 1 GOTO 1 ; saut à la feuille n° 1
IF      ()1000 = ()10 LENGTH 1 GOTO 2 ; saut à la feuille n°2
WRITEC  0 '/'
MESSAGE 102 SERIAL ABORT ; si feuille ni 1 ni 2 , lever une erreur
GOTO 199 ; passer toujours par le SEND après un message en ABORT
;
;
=====
;
;      FEUILLE N°1
;
=====
;
;      contrôle des marques horloges
;
LABEL 1 ;      contrôle pour la feuille n°1
CLOCK 50 1
IF      ()50 # ()3 LENGTH 3 GOTO 13 ; si nombre différent de 30 lever une erreur
;
;
=====
;
;      décodage du jour
;
IFERROR 104 SERIAL 3F 0 0 0 0 0 0 CONTINUE
FIELD 1002 1 1 35 1 1 1 3F HOR LEFT 0 1 1 4 1;travail en adresse fixe( pour
;exemple)
FIELD 1003 1 2 35 1 1 1 3F HOR LEFT 0 1 1 0 1
WRITEC  0 '/' ; le / est un séparateur de champs
;
;
=====
;
;      décodage du mois
;
;
IFERROR 105 SERIAL 3F 0 0 0 0 0 0 CONTINUE ; travail en adresse flottante
FIELD 0 3 1 23 1 1 3 3F HOR LEFT 0 1 1 0 1
WRITEC  0 '/'
;
=====

```

APPENDICE F : EXEMPLE DE PROGRAMMATION

```

;                                décodage de l'année
IFERROR 106 SERIAL 3F 1 0 0 0 0 0 CONTINUE ; avec already validé
FIELD 0 2 1 9 1 1 4 FF HOR LEFT 0 0 1 3 1 ; le pointeur bouge sans écriture en
;                                cas de non cochage
MEMORY 54 - 2 ; écriture au même endroit pour le contrôle already
FIELD 0 2 2 9 1 1 4 FF HOR LEFT 0 0 1 3 5
WRITEC 0 '/'

```

```

=====
;
;                                ;zone code à barre
GOSUB 16 ; sous routine pour lecture code barre pour toutes les feuilles

```

```

WRITEC 0 '/'

```

```

=====
;
;                                zones palettes
;
IFERROR 107 SERIAL 20 0 0 0 0 0 0 CONTINUE
FIELD 0 4 3 35 2 1 5 20 HOR DOWN 0 0 12 0 1
WRITEC 0 '/'

```

```

=====
;
;                                Quantités livrées
;
IFERROR 108 SERIAL 20 0 0 0 0 0 0 CONTINUE
FIELD 0 4 5 28 5 2 5 20 HOR DOWN 0 0 12 0 1
WRITEC 0 '/'

```

```

=====
;
;                                kms
;
IFERROR 109 SERIAL 3F 0 0 0 0 0 0 CONTINUE
FIELD 0 1 15 34 4 1 1 20 VER RIGHT 0 1 1 0 1 ; zone rendue obligatoire
WRITEC 0 '/' ; avec des ? en cas de non lecture pour la correction
ONLINE

```

```

=====
;
;                                observations
;
IFERROR 111 SERIAL 3F 0 0 0 0 0 0 CONTINUE
FIELD 0 1 15 17 11 1 2 30 HOR DOWN 0 0 1 1 4

```

APPENDICE F : EXEMPLE DE PROGRAMMATION

```

;                               horloge
;
;
WRITEC  0 '/'
IFERROR 112 SERIAL 3F 0 0 0 0 0 0 CONTINUE
FIELDXY 0 2 15 8 5 1 6 20 VER RIGHT 0 0 1 5 1 ;permet un cochage dans un groupe
                                                ;dont les alvéoles ne sont pas contigües
FIELDXY 0 2 17 10 3 4 7 20 VER RIGHT 0 0 1 5 1
GOTO    199

=====;
FEUILLE N°2
=====L
ABEL 2
;                               contrôle des marques horloges
;
;
;
IFMEMORY 55 # 33 GOTO 13 ;contrôle des marques horloges avec la memory 55
=====;
;
;                               zone avec input
;
MESSAGE 114 SERIAL SENDTEXT
INPUT   TO 1001 MAX 15
WRITEC  1016 '/'          ; remettre une adresse fixe après l'INPUT ,pour éviter un
                           ; enregistrement de longueur variable si la saisie clavier fait
                           ; moins de 15 caractères
;
=====;
;                               lecture du code à barre
GOSUB 16
=====;
;
;                               zone produit
;
FIELD  50 2 15 34 1 1 9 2E HOR DOWN 0 0 5 9 1;  lecture des zones
FIELD  52 2 16 34 1 1 9 2E HOR DOWN 0 0 5 9 1;  repentir et réponse
;                               ces champs aurait pu être pris en 1 fois
;                               par nb de groupe = 2
;
MEMORY 100 = 1 ; initialisation du compteur type produit ( L M N O P Q R S )
;
;

```

APPENDICE F : EXEMPLE DE PROGRAMMATION

```

;
;
;
LABEL 3
  IF      (052 = 070) LENGTH 2 GOTO 4; si la zone repentir est vierge valider la zone
                                           ;réponse
  MEMORY  60 = 52 ; sinon valider la zone repentir , 1° index du IF à la position du ;
                                           repentir
  GOTO    5
  MEMORY  58 + 7 ; déplacement de 7 de l'index pour la copie de la portion
;                                     d'enregistrement qui se place sur BON

LABEL 4
  MEMORY  60 = 50; 1° index du IF à la position de la zone réponse
;
LABEL 5
  MEMORY  58 = 200 ; Initialisation de l'index source du COPY à la position du début ;
                                           de la zone technique ( T.BON )
;
  IF      (00 = 080) LENGTH 2 GOTO 7 ; test de la valeur lue en zone 50 ou 52
  IF      (00 = 082) LENGTH 2 GOTO 7
  MEMORY  58 + 7 incrémentation de la mémoire d'origine du COPY pour
;                                     passer à Bon
  IF      (00 = 084) LENGTH 2 GOTO 7
  MEMORY  58 + 7
  IF      (00 = 086) LENGTH 2 GOTO 7
  MEMORY  58 + 7
  IF      (00 = 088) LENGTH 2 GOTO 7
  IF      (00 = 070) LENGTH 2 GOTO 8
  IF      (00 = 090) LENGTH 2 GOTO 6 ; si la zone est = à 31 ( toutes les cases cochées)
;                                     voir si c'est la zone repentir
  GOTO    11 ; sinon lever une erreur de double marquage
;
LABEL 6
  IF      (052 = 070) LENGTH 2 GOTO 11; lever de l'erreur double marquage
  GOTO    8
;
LABEL 7
  WRITEC  0 '/'
  MEMORY  100 TO 0 LENGTH 1 ; mettre devant la réponse produit un index pour
;                                     indiquer à quel produit se rattache le cochage
;
  WRITEC  0 '/'
;

```

APPENDICE F : EXEMPLE DE PROGRAMMATION

```
COPY FROM 0 TO 0 LENGTH 7 ; copier dans l'enregistrement la valeur du  
; cochage  
;  
;  
LABEL 8  
IFMEMORY 3 > 22 GOTO 9 ; tester si la lecture est arrivée sur le produit O ( ligne 25)  
MEMORY 3 + 2 ; si non passer au produit du dessous  
MEMORY 1 = 50  
RFIELD  
MEMORY 1 = 52  
MEMORY 3 + 1  
RFIELD  
MEMORY 100 + 1 ; incrémenter l'index de produit  
GOTO 3  
  
LABEL 9 ; une fois arrivé au produit O remonter au produit P  
;  
IFMEMORY 4 = 19 GOTO 10 ; sortie de la zone produit  
MEMORY 4 - 15 ; sinon se positionner au produit P  
MEMORY 3 - 10  
MEMORY 1 = 50  
RFIELD  
MEMORY 1 = 52  
MEMORY 3 + 1  
RFIELD  
MEMORY 100 + 1  
GOTO 3 ; remonter à la lecture des produits  
=====;  
zone signature lue mais non envoyée  
  
LABEL 10  
IFERROR 117 SERIAL 3F 0 0 0 1 0 0 CONTINUE ; au moins un groupe marqué ; pour  
la signature sinon erreur ( car une signature peut ne pas recouvrir tous les groupes ; du  
champs  
;  
;  
FIELD 300 1 29 24 5 1 10 20 HOR DOWN 0 0 20 0 1  
=====;  
;  
FIN DU DOCUMENT  
;  
;  
;  
LABEL 199  
SEND 1000 0 ; transmission de la partie, d'enregistrement d'édition, voulue  
NEXTDOC ; appel du document suivant  
;
```


APPENDICE F : EXEMPLE DE PROGRAMMATION

```
;
;
=====;
partie sousroutines
=====;
LABEL 11
  MESSAGE 116 SERIAL ABORT      ; message double marquage
  GOTO 199
;
LABEL 12
  MEMORY 54 + 10
  MESSAGE 110 SERIAL CONTINUE ; erreur code à barre sur la feuille n°1
  RETURN ; retour à la lecture des zones après le code à barre
;
LABEL 13
  MESSAGE 101 SERIAL ABORT; nombre de marques horloges incorrect
  GOTO 199

LABEL 14
  MESSAGE 103 SERIAL ABORT; erreur sur la zone identificatrice
  GOTO 199
;
;
LABEL 16 ; lecture du code barre
  IFERROR 100 SERIAL 3F 0 0 0 0 0 0 CONTINUE
  WRITEC 0 '?????????'
  MEMORY 54 - 10
  BARCODE 0 1 N
  MEMORY 60 = MEMORY 54
  IF (00 = 01) LENGTH 10 GOSUB 12
RETURN
;
;
;
@DEFINITION ; début des tables de données
;

TABLES 8,7,*,/
T BON
BON
MOYEN
INSUF.
NON OB.
/
```

APPENDICE F : EXEMPLE DE PROGRAMMATION

*; table données chaîne , les * marquent les positions à ignorer pour la fonction FIELDXY*
 TABLES 7,2,*,/

45

*

*

*

*

*

*

*

*

30

15

*

*

*

*

/

; zone palettes

TABLEN 5

800,400,200,100,80,40,20,10,8,4,

2,1

;

;

; table numérique avec 1 alvéole sur deux, ignorées

;

;

TABLEN 9

1,,2,,4,,8,,16

;

;

TABLEA 10,*

111111111111111111

;

;

; 0 1 2 3 4 5 6 7 8 9

TABLEA 1,*

01234567890

;

;

; 8 4 2 1

TABLEN 2

8,4,2,1

APPENDICE F : EXEMPLE DE PROGRAMMATION

; *zone horloge*

TABLES 6,2,*,/

*

10

09

08

*

11

*

*

*

07

12

*

*

*

06

01

*

*

*

05

*

02

03

04

*

/

;

;

;

;

TABLES 4,2,*,/

94

*

95

*

96

*

97

/

;

;

APPENDICE F : EXEMPLE DE PROGRAMMATION

```
;                                mois
TABLES 3,3,*,/
jan
fev
mar
avr
mai
jui
jul
aou
sep
oct
nov
dec
/

;                                tables de messages
;
TABLEM 100,*
    erreur marques de cadrage
;
TABLEM 101,*
    nombre de marques horloges incorrectes
;
TABLEM 102,*
    zone identificatrice incorrecte
;
TABLEM 103,*
    zone identificatrice non programmée
;
TABLEM 104,*
    erreur jour
;
TABLEM 105,*
    erreur mois
;
TABLEM 106,*
    erreur année
;
TABLEM 107,*
    erreur palette
;
TABLEM 108,*
    erreur quantites livrées
```

APPENDICE F : EXEMPLE DE PROGRAMMATION

```
;
TABLEM 109,*
  erreur kms
;
TABLEM 110,*
  erreur barre code
;
TABLEM 111,*
  erreur observation
;
TABLEM 112,*
  erreur horloge
;
TABLEM 113,*
  double marquage
;
TABLEM 114,*
  $entrez le nom de la personne
;
TABLEM 115,*
  année obligatoire
;
TABLEM 116,*
  double marquage produit
;
TABLEM 117,*
  signature absente
```

```
;
@END
```