

PILOTER & VISUALISER

LCA 250

**Guide utilisateur du terminal
LCA 250**

AVERTISSEMENT

Les instructions de service, manuels et logiciels sont protégés par les droits d'auteur. Tous les droits sont réservés. Toute copie, duplication, traduction, transposition totale ou partielle n'est pas autorisée. La confection d'une copie de réserve pour la propre utilisation représente la seule exception.

SOMMAIRE

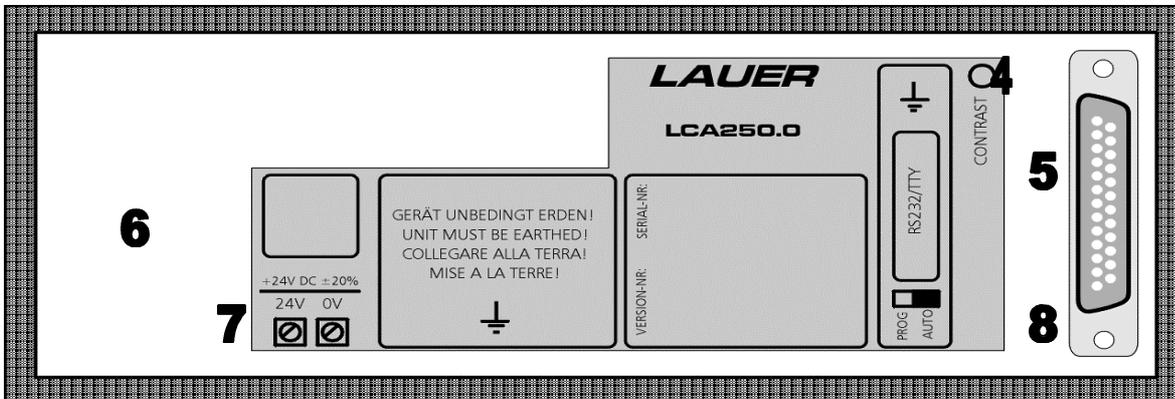
1. DESCRIPTION DU TERMINAL LCA 250	5
2. PRINCIPE DE FONCTIONNEMENT	6
3. EXPLOITATION DU TERMINAL LCA	8
3.1. Mode de fonctionnement	8
3.2. Gestion du clavier en mode PLC ou LCA	8
3.3. Lecture du code de la touche en mode PLC	8
3.4. Allumage et extinction des diodes de signalisation.....	9
3.5. Affichage des textes	10
3.5.1. Textes OPERATEUR.....	10
3.5.2. Textes MESSAGE.....	10
3.5.3. Textes d'AIDE	11
3.5.4. Textes MENU	11
3.6. Variables.....	13
3.6.1. Variables externes	13
3.6.1.1. Variable BIT	14
3.6.1.2. Variable STRING	15
3.6.1.3. Variable BCD.....	16
3.6.1.4. Variable BINAIRE.....	17
3.6.1.5. Variable ASCII.....	18
3.6.1.6. Variable WORD.....	18
3.6.1.7. Variable TIMER	19
3.6.2. Variables internes.....	20
3.7. Niveaux de priorité des textes à l'affichage.....	20
3.7.1. Liste des priorités	20
3.7.2. Validation des niveaux de priorité	21
3.8. Mémorisation et affichage des textes Message	21
3.9. Créer son propre caractère.....	22
4. LISTE D'AFFECTATION DES OCTETS MEMOIRE	23
5. DESCRIPTION DE LA TABLE MEMOIRE	24
5.1. Numéro du texte Menu, octet 0	24
5.2. Touches fonction, octet 1	24
5.3. Code des touches, octet 3	24
5.4. Code des touches, octet 4, 5	25
5.5. Sélection fichier, octet 6.....	25
5.6. Autorisation d'affichage, mémorisation des messages, octet 7	26
5.7. Textes Opérateurs, octet 9	26
5.8. Appel des textes Messages, octet 10 à 137.....	27
5.9. Variables, octet 138 à 255.....	27

6. ELEMENTS DE MISE EN OEUVRE	28
6.1. Diodes d'état.....	28
6.2. Mémorisation des fichiers dans le terminal LCA.....	29
6.3. Raccordements.....	30
6.3.1. Connecteur RS232 / TTY.....	30
6.3.2. Connecteur RS422 / RS485.....	30
6.3.3. Câble de programmation / simulation PCS 733.....	30
7. TABLE DES CARACTERES ASCII	31
8. CARACTERISTIQUES TECHNIQUES	32
9. AIDE A LA PROGRAMMATION	33
9.1. Fenêtre de configuration.....	33
9.2. Programmation d'une variable.....	34
9.3. Programmation d'un texte opérateur.....	36
9.4. Programmation d'un texte menu.....	37
9.5. Programmation d'un texte message.....	39
9.6. Programmation d'un texte d'aide.....	40
9.7. Programmation d'un caractère.....	41
9.8. Programmation des temps.....	41
9.9. Sauvegarde et transfert de l'application.....	42
9.10. Assistance technique.....	42
10. CONFIGURATION AUTOMATE	43
10.1. Automate AEG.....	43
10.2. Automate ALLEN-BRADLEY.....	45
10.3. Automate BOSCH.....	46
10.4. Automate FESTO.....	47
10.5. Automate GE FANUC.....	49
10.6. Automate HITACHI.....	50
10.7. Automate KEYENCE.....	51
10.8. Automate KLÖCKNER-MOELLER.....	52
10.9. Automate MITSUBISHI.....	54
10.10. Automate OMRON.....	55
10.11. Automate PHILIPS.....	57
10.12. Automate SIEMENS S5.....	58
10.13. Automate SIEMENS S7.....	60
10.14. Automate TELEMECANIQUE.....	61

1. DESCRIPTION DU TERMINAL LCA 250



- | | |
|--|--|
| <p>1 Afficheur LCD rétro-éclairé,
2 lignes de 40 caractères.</p> <p>2 7 touches de fonction libres de sérigraphie,
1 diode verte par touche.</p> <p>3 3 diodes d'état indiquant le mode de fonctionnement
du terminal LCA.</p> <p>4 Réglage du contraste de l'afficheur.</p> | <p>5 Connecteur liaison série RS232 / TTY.</p> <p>6 Connecteur liaison série RS422 / RS485.</p> <p>7 Bornes d'alimentation 24 VDC.</p> <p>8 1 fusible de protection.</p> |
|--|--|



Le cavalier AUTO/PROG doit toujours rester sur la position AUTO même durant le transfert du programme PC vers API.

2. PRINCIPE DE FONCTIONNEMENT

Le terminal **LCA 250** est spécialement conçu pour fonctionner avec un automate programmable.

L'échange des informations entre le terminal LCA et l'automate se fait au travers d'une **table mémoire** de 256 octets.

La lecture de cette table mémoire par le terminal LCA va lui permettre de savoir quels ordres ou informations lui sont transmis par l'automate :

- * **numéros des textes à afficher,**
- * **valeurs des variables,**
- * **états des voyants de signalisation, ...**

Inversement, la lecture de cette même table par l'automate va lui permettre de savoir quels ordres ou informations lui sont transmis par le terminal LCA :

- * **valeurs des consignes,**
- * **états des touches du clavier, ...**

Plus de 2000 fonctions sont ainsi ajoutées à l'automate, par la simple mise à jour de cette table mémoire et cela de façon totalement transparente pour votre programme automate.

TABLE MEMOIRE

octet 0	7	6	5	4	3	2	1	0
	128	64	32	16	8	4	2	1
	Numéro texte Menu actif (1...255)							
octet 1	7	6	5	4	3	2	1	0
	F 1	F 2	F 3	F 4	F 5	F 6	F 7	Présence LCA
	Touches fonction							
octet 2	7	6	5	4	3	2	1	0
octet 3	7	6	5	4	3	2	1	0
	LCA/PLC Mode	LCA/PLC Touche				4	2	1
	Code touche (1...7)							
octet 4	7	6	5	4	3	2	1	0
	L 1	L 2	L 3	L 4	L 5	L 6	L 7	
	Diodes VERTES fixes							
octet 5	7	6	5	4	3	2	1	0
	L 1	L 2	L 3	L 4	L 5	L 6	L 7	
	Diodes VERTES clignotantes							
octet 6	7	6	5	4	3	2	1	0
	Sélection Fichier							
octet 7	7	6	5	4	3	2	1	0
	D Autorisation affichage	I	I/D Mode de mémorisation	I/D	Menu	PLC	LCA	CONS.
	Autorisation							
octet 8	7	6	5	4	3	2	1	0
octet 9	7	6	5	4	3	2	1	0
	128	64	32	16	8	4	2	1
	Appel des textes Opérateur							
octet 10	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
	Appel des textes MESSAGE							
octet 11	7	6	5	4	3	2	1	0
	7	6	5	4	3	2	1	0
	Appel des textes MESSAGE							
à								
octet 136	7	6	5	4	3	2	1	0
	1023	1022	1021	1020	1019	1018	1017	1016
	Appel des textes MESSAGE							
octet 137	7	6	5	4	3	2	1	0
	1015	1014	1013	1012	1011	1010	1009	1008
	Appel des textes MESSAGE							
octet 138	7	6	5	4	3	2	1	0
	Variables Bit, String, BCD, BIN, Word, ASCII, Timer							
à								
octet 255	7	6	5	4	3	2	1	0
	Variables Bit, String, BCD, BIN, Word, ASCII, Timer							

3. EXPLOITATION DU TERMINAL LCA

3.1. Mode de fonctionnement

Le clavier du terminal LCA 250 peut être utilisé dans 2 modes de fonctionnement sélectionnés par la touche PLC/LCA :

- le mode PLC permet le transfert de l'état logique et du code de chaque touche vers l'automate.
- le mode LCA permet d'activer le menu de modification de variables, d'appeler les textes d'Aide, de feuilleter les textes de plus de 2 lignes, d'acquitter le clignotement éventuel d'un texte Message.

La touche PLC/LCA permet de basculer d'un mode à l'autre :

- la diode "**PLC**" située à gauche des touches indique que le mode PLC est actif.
- la diode "**LCA**" située à gauche des touches indique que le mode LCA est actif.

Nota :

La sélection du mode PLC/LCA peut être interdite à l'utilisateur par les bits de la table mémoire. Ces bits permettent également à l'automate de forcer le mode LCA ou PLC.

3.2. Gestion du clavier en mode PLC ou LCA

L'ensemble du clavier peut être traité comme de simples entrées automate.

L'état logique de chaque touche (appuyée ou relâchée) est donné par le bit associé dans la table mémoire quel que soit le mode sélectionné ; PLC ou LCA.

L'appui sur l'une des touches du clavier entraîne la mise à 1 du bit correspondant dans la table mémoire. Le relâchement de la touche remet ce bit à 0.

3.3. Lecture du code de la touche en mode PLC

A chaque appui d'une touche fonction un code de touche est transféré vers l'automate. Ce code permet d'identifier la dernière touche actionnée.

Le code binaire de la touche est transféré lors de l'appui de la touche et reste valide jusqu'à l'appui d'une autre touche.

3.4. Allumage et extinction des diodes de signalisation

Associées aux touches de fonction, les diodes de signalisation permettent de guider l'opérateur en lui signalant les touches autorisées, les actions à confirmer, ...

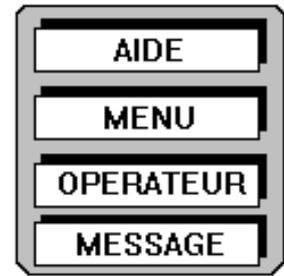
A chaque touche de fonction est associée 1 diode verte de signalisation. Les diodes sont à traiter comme de simples sorties automate, l'état de fonctionnement de chaque diode (éteint, allumé, clignotant court, clignotant long) est programmé dans les bits de poids identique.

Nota :

La diode d'état "**ERR**" située à gauche du terminal LCA signale une erreur de communication entre le terminal LCA et l'automate.

3.5. Affichage des textes

Pour indiquer clairement à l'opérateur le fonctionnement de la machine : données de production, messages d'alarmes, messages de défauts... le terminal LCA met à votre disposition différents types de messages dont la priorité à l'affichage est gérée par le terminal lui-même.



Les textes apparaissent à l'opérateur en fonction de leur degré d'importance et d'urgence.

Liste des différents types de textes :

Textes OPERATEUR	Priorité basse ↓ ↓ ↓
Textes MESSAGE -- d' INDICATION -- de DEFAUT	
Textes Menus	
Textes d' AIDE	Priorité haute

3.5.1. Textes OPERATEUR

Ils servent à indiquer à l'opérateur le fonctionnement normal de la machine. Ils sont au nombre de 256 et sont programmables sur 2 lignes de 40 caractères.

Nota :

Le texte Opérateur n° 000 est affiché à la mise sous tension du terminal LCA 250.

3.5.2. Textes MESSAGE

Ils servent à informer l'opérateur du fonctionnement de la machine. Selon le niveau d'importance, 2 types de Messages de priorité différente pourront être affichés.

L'affichage des textes Message pour chaque catégorie, Indication - Défaut, est tributaire d'une autorisation d'affichage. (Voir le chapitre NIVEAUX DE PRIORITE DES TEXTES A L'AFFICHAGE)

Les textes Message sont au nombre de 1024 et sont programmables de 1 à 32 lignes de 40 caractères. Le terminal LCA 250 affiche toujours les 2 premières lignes du message. L'opérateur peut appeler, en mode LCA, les lignes suivantes grâce à la touche [↓], une action sur la touche [↑] permet de revenir aux premières lignes.

L'affectation de chaque texte Message dans l'une des 2 catégories, Indication - Défaut, se fait de façon individuelle par programmation.

Les textes Messages peuvent être programmés clignotant et ce jusqu'à acquittement par l'opérateur à l'aide de la touche [↵] du terminal LCA.

Les textes Message sont appelés en positionnant à 1 le bit correspondant au numéro du message, l'appel des messages peut ainsi être comparé au positionnement d'une sortie automate.

3.5.3. Textes d'AIDE

Ils sont employés en tant que textes d'Aide à l'opérateur et sont affichés, en mode LCA, lorsque la touche [HELP] est appuyée.

Il existe un texte d'Aide pour chaque texte Opérateur, texte Menu, texte Message ils sont programmables de 2 à 32 lignes de 40 caractères.

Le terminal LCA 250 affiche toujours les 2 premières lignes du message. L'opérateur peut appeler les lignes suivantes grâce à la touche [↓], une action sur la touche [↑] permet de revenir aux premières lignes, la touche [HELP] restant appuyée.

3.5.4. Textes MENU

Ils sont utilisés pour constituer les différentes étapes du menu servant à paramétrer la machine. Ils sont au nombre de 256 et sont programmables sur 2 lignes de 40 caractères pour le terminal LCA 250.

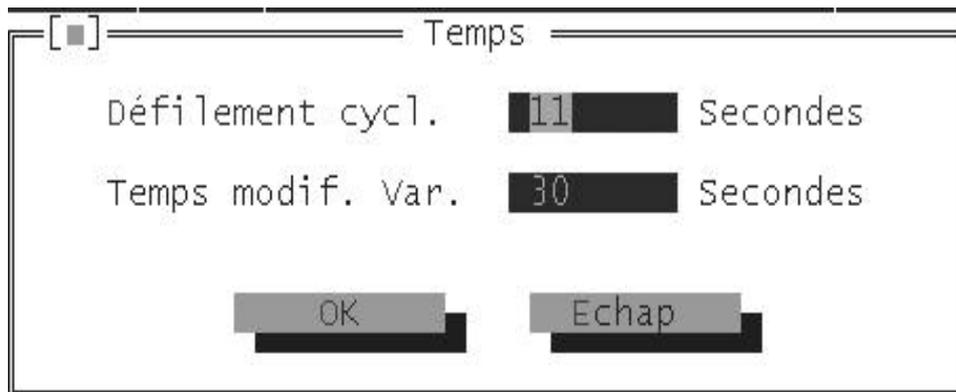
Le Menu est appelé, en mode LCA, par la touche [EDIT] du terminal LCA 250. L'opérateur peut alors à l'aide des touches [→] et [←] se déplacer dans le Menu. La touche [↑] permet d'aller au premier texte Menu et la touche [↓] permet d'accéder au dernier texte Menu. Une nouvelle action de la touche [EDIT] permet de sortir du Menu.

Nota 1 :

L'accès au menu peut être verrouillé par un bit de la table mémoire. La modification des variables consigne, dans les textes Menu, peut être interdite par un bit de la table mémoire.

Nota 2 :

Si dans le Menu, aucune touche n'est actionnée pendant un temps programmable, le Menu est automatiquement désactivé. Ce temps est défini à l'aide du logiciel LCAPRO dans la fenêtre "Projet" - "Temps". Lorsque ce temps est nul, la sortie automatique du Menu est désactivée.



Modification des variables dans le Menu

- 1° Appeler le Menu, en mode **LCA**, par la touche [EDIT].
- 2° Sélectionner le texte Menu par les touches [→] et [←] et le valider par la touche [↵]. Le curseur clignote alors sur la première variable du texte.
- 3° Se déplacer de variable en variable par les touches [→] et [←]. La modification de la variable s'effectue alors par incrémentation ou décrémentation de la valeur à l'aide des touches [↑] et [↓]. La validation de la nouvelle valeur de la variable s'effectue par la touche [↵].
- 4° Si la valeur de la variable n'est pas conforme à la programmation effectuée dans le terminal LCA, sa valeur est remplacée par des carrés noirs à l'affichage.
- 5° Une valeur non validée par la touche [↵] sera quand même prise en compte dès que l'une des touches de déplacement est actionnée ou que le Menu est désactivé.
- 6° Pour sélectionner à nouveau un texte Menu (comme au 1°), déplacer le curseur sur la dernière variable et appuyer sur la touche [↵].

Pour les textes menus il est possible de programmer, grâce au menu <<PROJET>> et par la commande Firmware, le texte devant être affiché lors de l'appel du menu par la touche [EDIT] du terminal LCA 250.



Il y a 3 possibilités :

L'opérateur se retrouve sur le dernier texte menu appelé ou toujours sur le premier texte menu ou toujours sur le dernier texte menu.

La commande "caractères du curseur" permet de programmer le caractère que l'opérateur aura à l'affichage au moment de la saisie des variables. La valeur de ce caractère est donnée en décimale.

La commande "Facteur de modification" permet, lors d'une modification de variable, de programmer la vitesse d'incrémentatation ou de décrémentation de la variable. Cette valeur est programmable de 0 à 9.

La valeur 0 étant la vitesse normale de modification des variables.

La valeur 1 étant la vitesse la plus lente et 9 la vitesse la plus rapide.

3.6. Variables

Pour être complet un message se doit d'afficher également les consignes et les paramètres de fonctionnement. Pour ce faire, le terminal LCA met à votre disposition des **variables externes** et **variables internes**.

Les variables externes sont les paramètres et les données dont dispose l'automate ; ces variables sont mémorisées dans la table mémoire. L'insertion de ces variables dans les messages est effectuée automatiquement par le terminal LCA.

Les variables internes sont des informations mémorisées dans le terminal LCA, c'est lui qui se charge de les réactualiser. Les variables internes sont du type : nombres de textes Message mémorisés par catégorie de priorité, numéro du message à l'affichage, ...

Chaque ligne des différents textes à afficher peut comporter jusqu'à 40 variables.

Elles peuvent être définies comme des :

- * **valeurs courantes** : elles ne sont pas modifiables par l'opérateur et sont transmises par l'automate au terminal LCA.
- * **valeurs de consigne** : elles sont modifiables par l'opérateur dans les textes Menu. Les données, une fois introduites à partir des touches du terminal LCA, sont transmises à l'automate.

3.6.1. Variables externes

Ces variables sont mémorisées dans les octets 138 à 255 de la table mémoire.

7 formats de variables sont disponibles : **BIT, STRING, BCD, BIN, ASCII, WORD, TIMER**

Les variables peuvent être des données numériques ou des textes

Nota 1 :

Une variable peut être programmée à une adresse de texte Message

3.6.1.2. Variable STRING

Les variables STRING sont identifiables à des commutateurs à maxi 256 positions.

Pour chaque valeur 0 à 255 de la variable, le terminal LCA affichera le texte alphanumérique associé. Ces textes associés, de maximum 40 caractères, sont mémorisés dans le terminal LCA.



Un octet mémoire est nécessaire pour mémoriser une telle variable.

Le changement d'état d'une variable STRING lorsqu'elle est définie en tant que CONSIGNE, est effectué par l'opérateur dans le **Menu** à l'aide des touches [↓] et [↑] du terminal LCA. Le changement de la variable est à confirmer par la touche [↵].

Exemple :

L'installation fonctionne en mode : MANUEL
 _____/ _____/
 Message fixe Variable STRING, valeur 0 = 0000 0000

L'installation fonctionne en mode : AUTOMATIQUE
 _____/ _____/
 Message fixe Variable STRING, valeur 1 = 0000 0001

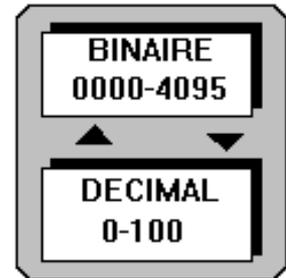
L'installation fonctionne en mode : PAS A PAS
 _____/ _____/
 Message fixe Variable STRING, valeur 2 = 0000 0010

L'installation fonctionne en mode : REGLAGE
 _____/ _____/
 Message fixe Variable STRING, valeur 3 = 0000 0011

3.6.1.4. Variable BINAIRE

Les variables BIN permettent l'affichage ou l'introduction de données directement mémorisées en code binaire dans la table mémoire de l'automate.

Il va de soit que l'affichage et l'introduction de ces variables se font de façon décimale.



Il est possible de définir les valeurs minimum et maximum que peut prendre chaque variable BIN.

Si l'opérateur introduit une valeur hors tolérances, le terminal LCA affiche automatiquement :

- * la valeur minimum si la valeur introduite est inférieure au seuil programmé.
- * la valeur maximum si la valeur introduite est supérieure au seuil programmé.

La modification d'une variable BIN définie en tant que CONSIGNE est effectuée par l'opérateur dans le **Menu** à l'aide des touches [↓] et [↑] du terminal LCA. Le changement de la variable est à confirmer par la touche [↵].

Les tests effectués lors de la modification d'une variable sont entièrement gérés par le terminal LCA. L'automate est entièrement déchargé de ces tâches, il est sûr de trouver la bonne valeur dans la table mémoire.

Il est également possible de réaliser une conversion d'échelle entre la valeur introduite au clavier et la valeur mémorisée dans la table mémoire de l'automate.

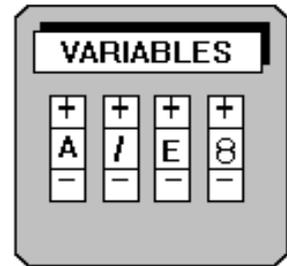
Cette mise à l'échelle est validée par la commande "[] mise à l'échelle" lors de l'édition de la variable.

Exemple :

Température d'extrusion : +152.6 °C
 └──┬──┘
 Variable VBIN-WORD

3.6.1.5. Variable ASCII

Les variables ASCII permettent l'affichage ou l'introduction de données directement mémorisées en code ASCII dans la table mémoire de l'automate.



La longueur de la variable ASCII est de 1 caractère.

Chaque caractère nécessite un octet mémoire (8 bits).

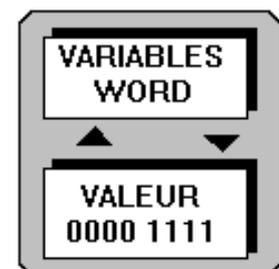
Le changement d'état de chaque caractère d'une variable ASCII, lorsqu'elle est définie en tant que CONSIGNE, est effectué par l'opérateur dans le **Menu** à l'aide des touches [↓] et [↑] du terminal LCA. Le changement de la variable est à confirmer par la touche [↵].

Exemple :

Nom de l'opérateur : MARTIN
└──────────┘
 6 Variables ASCII

3.6.1.6. Variable WORD

Les variables WORD permettent l'affichage et la modification de deux octets de la table mémoire dans différents formats.



KH: format hexadécimal. Une telle variable occupe 4 digits à l'affichage.

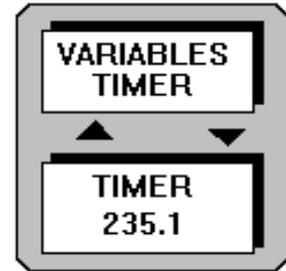
KM: format binaire. Une telle variable occupe 17 digits à l'affichage.

KY: format décimal pour chaque octet. Une telle variable occupe 7 digits à l'affichage.

Le changement d'état d'une variable WORD, lorsqu'elle est définie en tant que CONSIGNE, peut être effectué par l'opérateur dans le **Menu** à l'aide des touches [↓] et [↑] du terminal LCA. Le changement de la variable est à confirmer par la touche [↵].

3.6.1.7. Variable TIMER

Les variables **TIMER** permettent l'affichage ou l'introduction de données directement mémorisées dans la table mémoire de l'automate dans un code spécifique aux automates Siemens.



Cette variable utilise 2 octets mémoire et se présente de la façon suivante dans la table mémoire :

« 00dd cccc bbbb aaaa »

- aaaa = valeur BCD (0...9) unité du mot
- bbbb = valeur BCD (0...9) dizaine du mot
- cccc = valeur BCD (0...9) centaine du mot
- dd = valeur de la base de temps du mot (0...3)

A l'affichage cette variable prend 3 digits pour les unités, dizaines, centaines puis est suivie d'un texte associé de maximum 37 caractères pour chaque valeur de la base de temps (00, 01, 10, 11).

Le changement d'état d'une variable **TIMER**, lorsqu'elle est définie en tant que **CONSIGNE**, peut être effectué par l'opérateur dans le **Menu** à l'aide des touches [↓] et [↑] du terminal LCA. Le changement de la variable est à confirmer par la touche [↵].

Exemple :

Modification de la temporisation 1: 452 * 10 milli-secondes



- 452** : valeur BCD de la temporisation (cccc bbbb aaaa)
- * 10 milli-secondes** : texte associé à la valeur de la base de temps (dd)

3.6.2. Variables internes

Le terminal LCA met à votre disposition 4 variables internes dont l'affectation est prédéfinie.
Ces variables peuvent être insérées dans les différents messages programmés dans le terminal LCA.

NOM	DESCRIPTION	LG	TYPE
[INDICATION]	Nb. de Messages INDICATION en mémoire	4	courante
[DEFAULT]	Nb. de Messages DEFAULT en mémoire	4	courante
[NUMERO_TEXTE]	Numéro du texte appelé	4	courante
[MESSAGE]	Nb. de Messages appelés	4	courante

3.7. Niveaux de priorité des textes à l'affichage

3.7.1. Liste des priorités

Le terminal LCA permet de gérer 5 niveaux de priorité.
L'affichage des différents textes est hiérarchisé par une gestion de priorité interne au terminal LCA.

Liste des différents types de textes et de leur niveau de priorité :

MESSAGES	NIVEAU	PRIORITE
Textes Opérateur	0	Faible
Textes Message - Catégorie Indication I	1	
Textes Message - Catégorie Défaut D	2	
Textes Menu M	3	↓
Textes d'Aide	4	Forte

Il est nécessaire d'affecter, lors de la programmation, à chaque texte Message une catégorie de priorité en fonction de l'importance de son contenu.

Les textes d'Aide sont prioritaires sur tous les autres textes. Ils sont affichés, en mode LCA uniquement, lorsque la touche [HELP] est activée.

3.7.2. Validation des niveaux de priorité

Le terminal LCA affiche toujours le texte ayant la priorité la plus élevée.

Il est néanmoins possible d'afficher des textes de priorité inférieure, grâce aux bits de commande de la table mémoire.

La priorité d'affichage des textes d'Aide ne peut être dévalidée. En effet, le texte d'Aide appelé lors de l'appui de la touche [HELP] dépend du texte actuellement à l'affichage.

A chaque **texte Opérateur, texte Menu, texte Message d'Indication, texte Message de Défaut** est associé un texte d'Aide spécifique.

3.8. Mémorisation et affichage des textes Message

Nous avons vu, dans le chapitre *AFFICHAGE DES TEXTES - Textes Message*, que l'appel d'un texte Message se fait en positionnant à 1 le bit correspondant au numéro du message.

Lorsque l'opérateur doit être informé de plusieurs anomalies simultanées, il est nécessaire d'appeler plusieurs textes Message en même temps. Le terminal LCA mémorise les 128 derniers textes Message, leur ordre d'affichage dépend du mode de mémorisation programmé.

Pour ce faire, le terminal LCA met à votre disposition 3 modes de mémorisation.

L'affectation d'un mode de mémorisation se fait par programmation.

PREMIER message avec scrutation :

Le plus ancien des textes appelés **dans sa classe de priorité** est affiché ; tous les autres textes appelés par la suite sont mémorisés. Après acquittement (ou effacement) de ce premier message, le second message sera affiché et ainsi de suite. La mémorisation "premier message" s'effectue donc du plus ancien message vers le plus récent.

L'opérateur peut en mode LCA visualiser les autres textes mémorisés grâce aux touches [→] et [←].

DERNIER message avec scrutation :

Le dernier texte appelé dans sa classe de priorité est affiché ; tous les textes précédemment appelés restent mémorisés jusqu'à leur acquittement. Après acquittement (ou effacement) de ce dernier message, l'avant dernier message sera à nouveau affiché et ainsi de suite. La mémorisation du "dernier message" s'effectue donc du plus récent message vers le plus ancien.

L'opérateur peut en mode LCA visualiser les autres textes mémorisés grâce aux touches [→] et [←].

Message CYCLIQUE :

Tous les textes Message mémorisés sont affichés de façon cyclique dans leur classe de priorité. Le temps de défilement est programmable en secondes à l'aide du logiciel de programmation. Il est initialisé par Défaut à 10 secondes.

3.9. Créer son propre caractère

Le terminal LCA autorise la création de 8 caractères spécifiques qui n'existe pas dans la table ASCII. Ces 8 caractères sont intégrés dans la table des caractères ASCII aux adresses 08h à 0Fh.

Un caractère se compose d'une matrice de 5 x 8 points (8 lignes à 5 colonnes).

	Colonne				
	1	2	3	4	5
Ligne 1	■	■	■	■	■
Ligne 2	■	■	■	■	■
Ligne 3	■	■	■	■	■
Ligne 4	■	■	■	■	■
Ligne 5	■	■	■	■	■
Ligne 6	■	■	■	■	■
Ligne 7	■	■	■	■	■
Ligne 8	■	■	■	■	■

Un caractère spécifique se compose en allumant ou non les différents points de la matrice par programmation.

Exemple :

Créer le caractère flèche vers le bas

	Colonne				
	1	2	3	4	5
Ligne 1			■		
Ligne 2			■		
Ligne 3			■		
Ligne 4			■		
Ligne 5			■		
Ligne 6	■		■		■
Ligne 7		■	■	■	
Ligne 8			■		

4. LISTE D'AFFECTION DES OCTETS MEMOIRE

La liste d'affectation permet d'affecter l'adresse de chaque octet de la table mémoire du terminal LCA à une adresse quelconque dans la mémoire de l'automate.

Cette liste d'affectation se programme, à l'aide du logiciel de programmation, dans la fenêtre 'EDITION - LISTE D'AFFECTION'.

La programmation peut être réalisée octet par octet ou affectée à toute la table mémoire, pour cela il suffit de mettre une croix dans la commande 'Zone d'affectation' en y spécifiant le nombre de mot.

La zone des 256 octets mémoire n'a pas besoin d'être consécutive.

Les octets de la table mémoire peuvent être directement affectés à des sorties, entrées, bits internes, mots internes de l'automate, ...

Exemple pour un automate Siemens :

Les touches de fonction peuvent directement commander des sorties automate (AB1).

Les entrées de l'automate peuvent directement appeler des messages... (MB25, DB50 DW21..)

Textes messages 0 à 7 sont appelés par l'automate dans le MB1

Touches de fonction F1 à F7 commandent directement des sorties AB2

Les leds associées aux touches de fonction sont appelées par le DW5 du DB21...

5. DESCRIPTION DE LA TABLE MEMOIRE

5.1. Numéro du texte Menu, octet 0

	7	6	5	4	3	2	1	0
octet 0	128	64	32	16	8	4	2	1
	Numéro texte Menu actif (1...255)							

L'octet 0 permet de signaler à l'automate que l'opérateur a appelé les textes Menus en indiquant le numéro du texte affiché.

5.2. Touches fonction, octet 1

	7	6	5	4	3	2	1	0
octet 1	F 1	F 2	F 3	F 4	F 5	F 6	F 7	Présence LCA
	Touches fonction							

Bit 0 Le bit 0 permet de vérifier la communication entre le terminal LCA et l'automate. Ce bit est cycliquement mis à 1 par le terminal LCA.

Bit 1 à 7 L'appui sur l'une des touches du clavier entraîne la mise à 1 d'un bit de la table mémoire. Le relâchement de la touche remet ce bit à 0.

5.3. Code des touches, octet 3

	7	6	5	4	3	2	1	0
octet 3	LCA/PLC Mode	LCA/PLC Touche				4	2	1
	Code touche (1...7)							

Bit 0 à 2 Permet d'identifier la dernière touche actionnée. Le code binaire de la touche est transféré lors de l'appui de la touche et reste valide jusqu'à l'appui d'une autre touche.

Touche	Bit 2	Bit 1	Bit 0
F1	0	0	1
F2	0	1	0
F3	0	1	1
F4	1	0	0
F5	1	0	1
F6	1	1	0
F7	1	1	1

Bit 6 L'appui sur la touche [PLC/LCA] entraîne la mise à 1 de ce bit. Le relâchement de la touche remet ce bit à 0.

Bit 7 Signale si l'opérateur travaille en mode LCA ou PLC
 A l'état 0 indique le mode PLC (transfert de l'état logique des touches vers l'automate)
 A l'état 1 indique le mode LCA (fonction interne au terminal LCA)

5.4. Diodes de signalisation, octet 4, 5

octet 4	7	6	5	4	3	2	1	0
	L 1	L 2	L 3	L 4	L 5	L 6	L 7	
Diodes VERTES fixes								
octet 5	7	6	5	4	3	2	1	0
	L 1	L 2	L 3	L 4	L 5	L 6	L 7	
Diodes VERTES clignotantes								

A chaque touche de fonction est associé une diode de signalisation. Les diodes sont à traiter comme de simples sorties automatés, l'état de fonctionnement de chaque diode (éteint, allumé, clignotement court, clignotement long) est programmé dans les octets 4 et 5.

Bit n°x de l'octet 4	Bit n°x de l'octet 5	DIODE
0	0	Eteinte
1	0	Allumée
0	1	Clignotante long
1	1	Clignotante court

5.5. Sélection fichier, octet 6

octet 6	7	6	5	4	3	2	1	0
								Sélection Fichier

Le terminal LCA permet de mémoriser 2 fichiers programme. Ces 2 fichiers peuvent être deux programmes différents ou 1 programme en 2 langues. La sélection du fichier de travail est réalisée par le bit 0 de l'octet 6.

Bit 0 A l'état 0 indique que c'est le fichier 1 qui est appelé
 A l'état 1 indique que c'est le fichier 2 qui est appelé

Nota 1 :

Lors de l'utilisation du fichier 2 la table d'affectation est identique au fichier 1.

5.6. Autorisation d'affichage, mémorisation des messages, octet 7

octet 7	7	6	5	4	3	2	1	0
	D Autorisation affichage	I	I/D Mode de mémorisation	I/D	Menu	PLC	LCA	CONS. Autorisation

- Bit 0 A l'état 0 autorise la modification des consignes
A l'état 1 n'autorise pas la modification des consignes
- Bit 1 A l'état 0 validation de la touche [PLC/LCA]
A l'état 1 verrouillage de la touche [PLC/LCA] en mode LCA (mode interne)
- Bit 2 A l'état 0 validation de la touche [PLC/LCA]
A l'état 1 verrouillage de la touche [PLC/LCA] en mode PLC
- Bit 3 A l'état 0 autorisation d'affichage des textes Menu
A l'état 1 Pas d'autorisation d'affichage des textes Menu
- Bit 4 et 5 Mode de mémorisation des textes Messages Indication Défaut.

L'affectation du mode de mémorisation se fait par programmation. Le terminal permet de mémoriser les 128 derniers messages Indication et les 128 derniers messages Défaut.

Bit 5	Bit 4	
0	0	Dernier message
0	1	Premier message
1	0	Message cyclique
1	1	Fonction réservée

- Bit 6 A l'état 0 autorisation d'affichage des textes Messages Indication
A l'état 1 pas d'autorisation d'affichage des textes Messages Indication
- Bit 7 A l'état 0 autorisation d'affichage des textes Messages Défaut
A l'état 1 pas d'autorisation d'affichage des textes Messages Défaut

5.7. Textes Opérateurs, octet 9

octet 9	7	6	5	4	3	2	1	0
	128	64	32	16	8	4	2	1

Appel des textes Opérateur

L'octet 9 permet d'appeler les textes Opérateurs de façon numérique (0 à 255).

5.8. Appel des textes Messages, octet 10 à 137

	7	6	5	4	3	2	1	0
octet 10	15	14	13	12	11	10	9	8
	Appel des textes MESSAGE							
	7	6	5	4	3	2	1	0
octet 11	7	6	5	4	3	2	1	0
	Appel des textes MESSAGE							
	à							
	7	6	5	4	3	2	1	0
octet 137	1015	1014	1013	1012	1011	1010	1009	1008
	Appel des textes MESSAGE							

Les textes Messages sont au nombre de 1024 et sont programmables de 1 à 32 lignes de 40 caractères. Les textes Messages sont appelés en positionnant à 1 le bit correspondant au numéro du message, l'appel des messages peut ainsi être comparé au positionnement d'une sortie automate.

Nota :

Les octets réservés aux textes Messages peuvent être utilisés pour les variables.

5.9. Variables, octet 138 à 255

	7	6	5	4	3	2	1	0
octet 138								
	Variables Bit, String, BCD, BIN, Word, ASCII, Timer							
	à							
	7	6	5	4	3	2	1	0
octet 255								
	Variables Bit, String, BCD, BIN, Word, ASCII, Timer							

Pour être complet un message se doit d'afficher également les consignes et les paramètres de fonctionnement. La zone des variables se situe entre l'octet 138 et 255.

Les variables peuvent également utiliser les octets des textes Messages.

6. ELEMENTS DE MISE EN OEUVRE

6.1. Diodes d'état

Ces 3 diodes indiquent le mode de fonctionnement du terminal LCA.

6.1.1. Diode verte "PLC" allumée :

L'état des touches F1 à F7 et le code de la touche appuyée sont transférés vers l'automate.

6.1.2. Diode verte LCA allumée :

L'état des touches F1 à F7 est transféré vers l'automate.

Les touches permettent de modifier également les variables dans les textes Menu, de visualiser les textes d'Aide, ou d'acquitter le clignotement des textes messages.

6.1.3. Diode rouge ERR :

Le terminal LCA signale une erreur de communication avec l'automate.

Diode ALLUMEE :

La communication avec l'automate n'a pas encore été établie depuis la mise sous tension du terminal LCA

Diode CLIGNOTANTE :

La communication avec l'automate a été interrompue, le terminal LCA affiche le message
===== COMMUNICATION ERROR =====.

6.2. Mémorisation des fichiers dans le terminal LCA

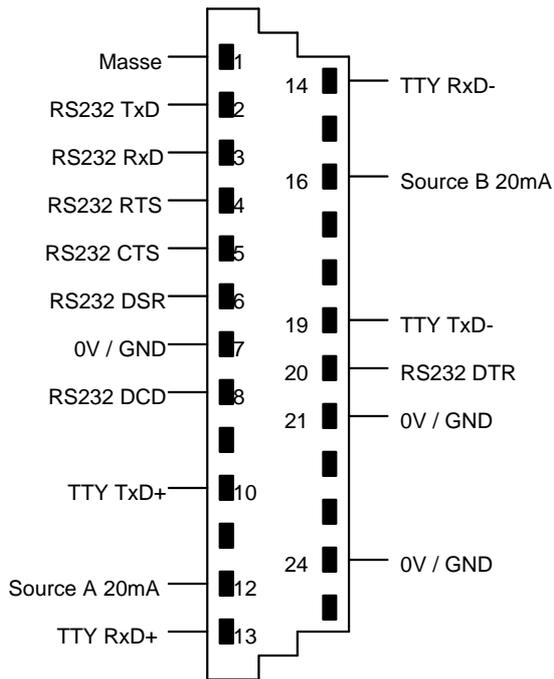
Le terminal LCA permet de mémoriser soit un fichier de 34 KB soit deux fichiers de 17 KB.

Cette sélection est réalisée lors de la compilation du programme avec le logiciel LCAPRO.

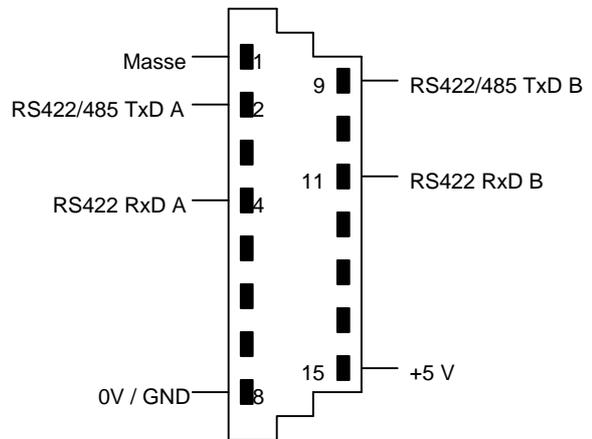
L'automate peut à tout moment, si deux fichiers sont programmés, basculer d'un fichier à l'autre grâce à un bit de commande.

6.3. Raccordements

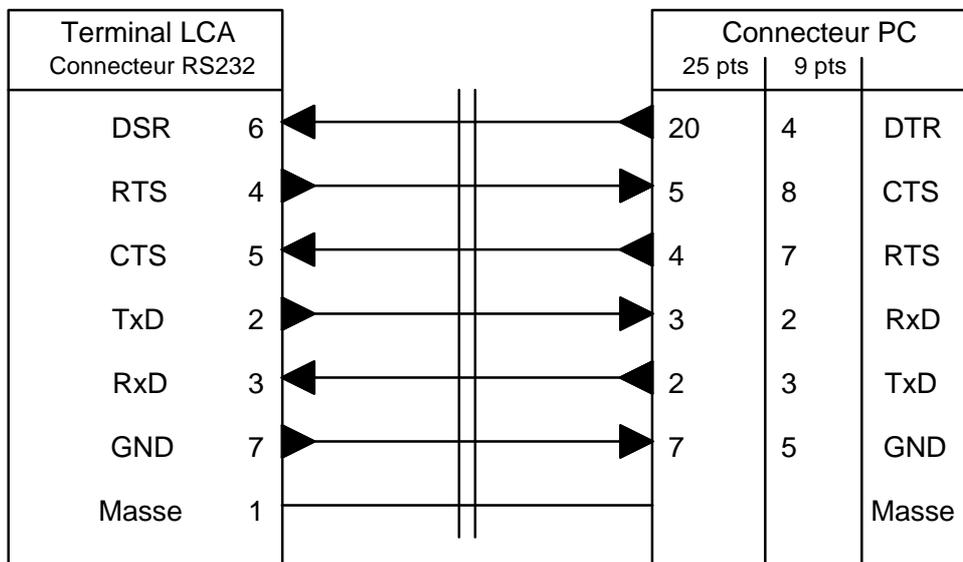
6.3.1. Connecteur RS232 / TTY



6.3.2. Connecteur RS422 / RS485



6.3.3. Câble de programmation / simulation PCS 733



7. TABLE DES CARACTERES ASCII

	▶		Ø	Ɔ	Ɔ	˘	Ɔ	Ɔ	α		o	À	Ɔ	à	ÿ
	◀	!	1	À	Q	Ɔ	Ɔ	Ɔ	Ɔ	i	±	Á	Ñ	á	ñ
	“	”	2	B	R	b	r	Ж	Г	Ɔ	z	Â	Ò	â	ò
	”	#	3	C	S	c	s	Э	π	Ɔ	z	Ã	Ó	ã	ó
	⌠	\$	4	D	T	d	t	И	Σ	×	Ɔ	Ä	Ô	ä	ô
	⌡	%	5	E	U	e	u	Й	σ	¥	μ	Å	Ö	å	ö
	●	&	6	F	V	f	v	Ј	Ј	!	¶	Æ	Ö	æ	ö
	◄	,	7	G	W	g	w	П	τ	§	•	Ç	×	ç	÷
(1)	↑	(8	H	X	h	x	У	•	Ɔ	ω	È	Ɔ	è	Ɔ
(2)	↓)	9	I	Y	i	y	Ц	Ө	Ө	¹	É	Ù	é	ù
(3)	→	*	:	J	Z	j	z	Ч	Ω	∞	Ω	Ê	Ú	ê	ú
(4)	←	+	;	K	[k	[Ш	δ	«	»	Ë	Û	ë	û
(5)	≤	,	<	L	\	l		Щ	∞	Ю	¼	Ì	Ü	ì	ü
(6)	≥	-	=	M]	m]	Ъ	•	Я	½	Í	Ý	í	ý
(7)	▲	.	>	N	^	n	~	Ы	ε	Ɔ	¼	Î	Ɔ	î	Ɔ
(8)	▼	/	?	O	_	o	ó	Э	Π	•	¿	Ï	Ɔ	ï	Ɔ

8. CARACTERISTIQUES TECHNIQUES

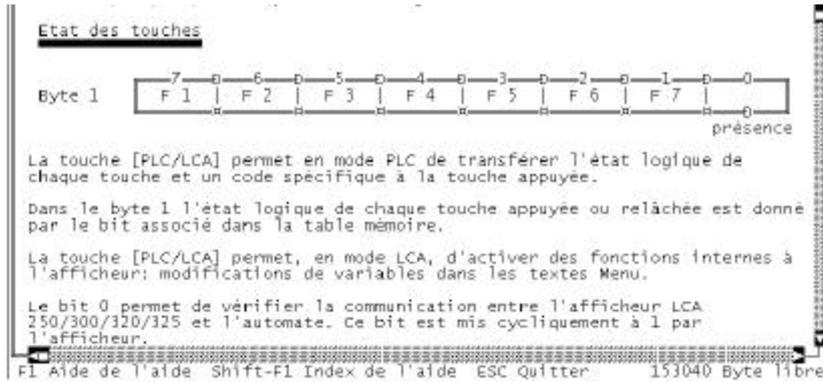
Tension d'alimentation	24 VDC \pm 20%
Consommation	8 W
Affichage	LCD rétro-éclairé
Hauteur des caractères	5 mm
Format des caractères	Matrice 5 x 8
Nombre de lignes d'affichage	2
Nombre de caractères par ligne	40
Jeu de caractères	International
Mémoire	Flash-EEPROM
Liaison série	RS232/TTY & RS422/RS485
Température de fonctionnement	0 ... +50°C
Température de stockage	-25 ... +70°C
Poids approximatif	450 g
Indice de protection face avant	IP 65
Indice de protection face arrière	IP 20
Dimensions (LxHxP)	216 x 84 x 57 mm
Découpe (LxH)	208 x 76 mm



Le cavalier AUTO/PROG doit toujours rester sur la position AUTO même durant le transfert du programme PC vers API.

**MATERIEL CERTIFIE
CONFORME NORME CE UL VDE**

Cette fenêtre permet la visualisation de toute la table mémoire par les touches montée et descente de votre PC. La touche F1 permet, à tout moment, d'appeler une aide en ligne.



Démarrer un nouveau projet par le menu <<EDITER>> et la commande "NOUVEAU"



9.2. Programmation d'une variable

Le Menu <<EDITER>> et la commande "Variable" permet de programmer les différentes variables.

Exemple de programmation d'une variable :



Une variable est définie par un nom de 16 caractères maximum. Les variables marqués par des [] sont des variables internes. Un ensemble de commandes permet de "copier", "effacer", "renommer" les variables. La commande "utilisé" réalise une liste croisée de la variable sélectionnée.

Exemple :

La commande "éditer" ouvre la fenêtre de programmation de la variable.



La définition d'une variable se réalise de la façon suivante :

- Adresse Donner une adresse de la table mémoire, le logiciel donne automatiquement la première adresse de libre de la table mémoire
- Format Donner un format pour la variable
- Type Donner un type de variable
- Minimum Permet de borner les variables binaires
- maximum
- Digits Donne le nombre de digit utilisé pour la variable

Exemple de la liste de toutes les variables utilisées pour le programme DEMO :



9.3. Programmation d'un texte opérateur

Le Menu <<EDITER>> et la commande "Texte Opérateur" permet de programmer les textes opérateurs.

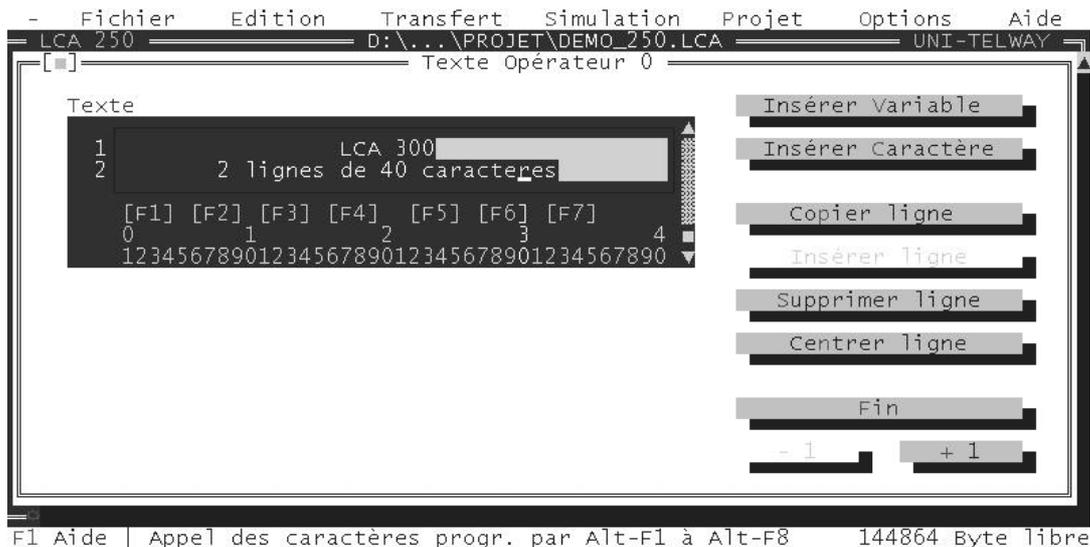
Exemple de programmation d'un texte opérateur :



Un texte opérateur est défini par un numéro. Le nombre maximum de textes opérateur est de 256 (0 à 255). Le texte opérateur 0 (texte de repos) est affiché à la mise sous tension.

Exemple :

La commande "Editer" ouvre la fenêtre de programmation du texte opérateur.



Un ensemble de commandes permet d'insérer une variable ou un caractère ; copier, insérer, supprimer, centrer une ligne, de passer au texte suivant ou au texte précédent.

Remarque :

Lorsqu'une variable est insérée sa longueur est représentée par des rectangles et sa définition est donnée lorsqu'on se place sur le premier caractère de la variable.

9.4. Programmation d'un texte menu

Le Menu <<EDITER>> et la commande "texte Menu..." permet de programmer les menus.

Exemple de programmation d'un texte menu :



Un menu se présente à l'affichage comme une succession de textes opérateur. Le menu est appelé, en mode LCA, par le terminal grâce à la touche [EDIT]

Un ensemble de commandes permet d'éditer, de copier, d'effacer, de renuméroter un menu.

Exemple :

La commande "Editer" ouvre la fenêtre de programmation d'un texte menu.



Un ensemble de commandes permet d'insérer une variable ou un caractère ; copier, insérer, supprimer, centrer une ligne ; de passer au texte suivant ou au texte précédent.

Exemple de la liste des textes menus utilisés pour le programme DEMO.



Pour les textes menus il est possible de programmer, grâce au menu <<PROJET>> et par la commande Firmware, le texte devant être affiché lorsque l'opérateur appelle le menu par la touche [EDIT] du terminal LCA 250.



Il y a 3 possibilités :

L'opérateur se retrouve sur le dernier texte menu appelé ou toujours sur le premier texte menu ou toujours sur le dernier texte menu.

La commande "caractères du curseur" permet de programmer le caractère que l'opérateur aura à l'affichage au moment de la saisie des variables. La valeur de ce caractère est donnée en décimale.

9.5. Programmation d'un texte message

Le Menu <<EDITER>> et la commande "Texte Message..." permet de programmer les textes messages.

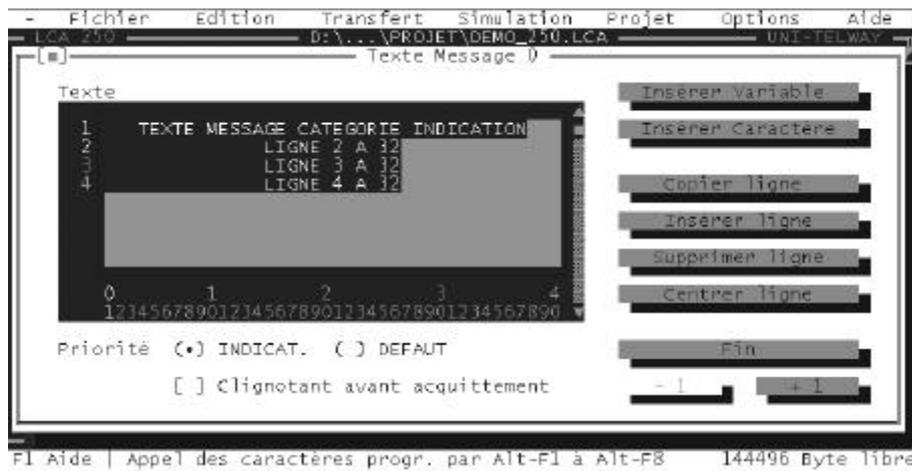
Exemple de programmation d'un texte message :



Un texte message est défini par un numéro allant de 0 à 1023.

Exemple :

La commande "Editer" ouvre la fenêtre de programmation du texte message.



Un texte Message peut contenir 32 lignes. Les textes messages sont classés en 2 priorités (indication, défaut) et peuvent être programmés en mode clignotant avant acquittement.

Un ensemble de commandes permet d'insérer une variable ou un caractère ; copier, insérer, supprimer, centrer une ligne, de passer au texte suivant ou au texte précédent.

La liste des textes messages permet de visualiser tous les textes déjà programmés et leur priorité.

I = indication

D = défaut



9.6. Programmation d'un texte d'aide

Le Menu <<EDITER>> et la commande "Texte d'aide.." permet de programmer les textes d'aides.

Exemple de programmation d'un texte d'aide :



Il existe un texte d'aide pour chaque texte. Il faut sélectionner la catégorie du texte puis le numéro du texte d'aide. Ce texte d'aide sera automatiquement appelé par l'opérateur, en mode LCA, par la touche [HLP].

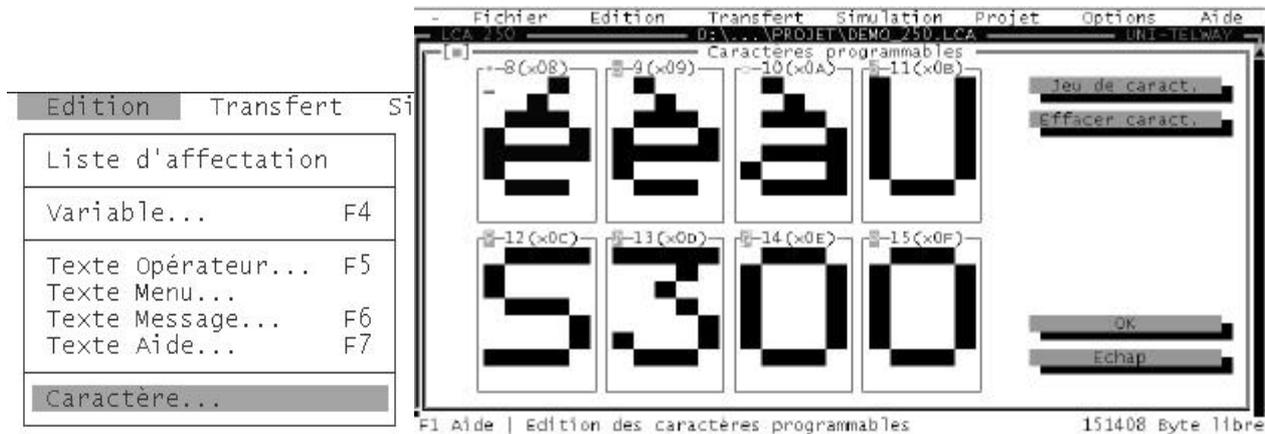
Un texte d'aide peut contenir de 2 à 32 lignes.

Un ensemble de commandes permet d'insérer une variable ou un caractère ; copier, insérer, supprimer, centrer une ligne, de passer au texte suivant ou au texte précédent.

9.7. Programmation d'un caractère

Le Menu <<EDITER>> et la commande "Caractère" permet de programmer des caractères non représentés dans la table ASCII.

Exemple de programmation d'un caractère :



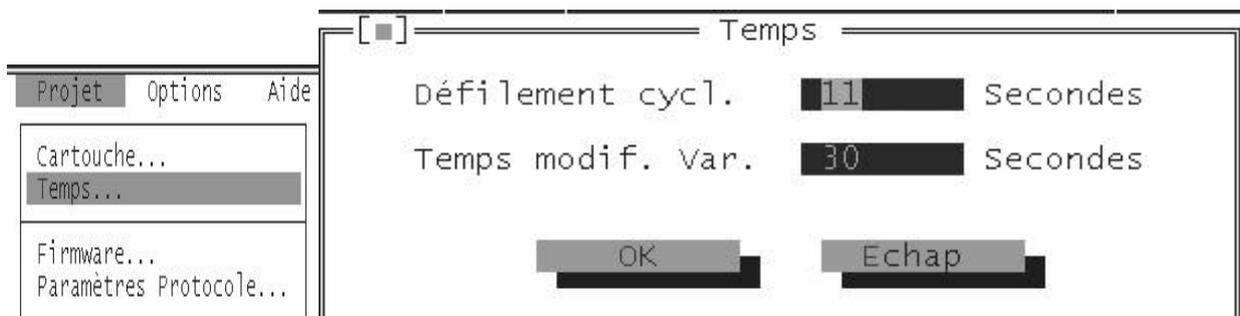
Les caractères programmables sont aux nombres de 8.

Une action sur la touche "SPACE" permet de valider ou pas un point de la matrice. Les touches de direction permettent de passer d'un point à l'autre.

9.8. Programmation des temps

Le Menu <<PROJET>> et la commande "Temps" permet également de programmer le temps de défilement cyclique lors de l'appel de plusieurs textes messages et le temps d'attente maximum avant de quitter les textes menus.

Exemple de programmation :



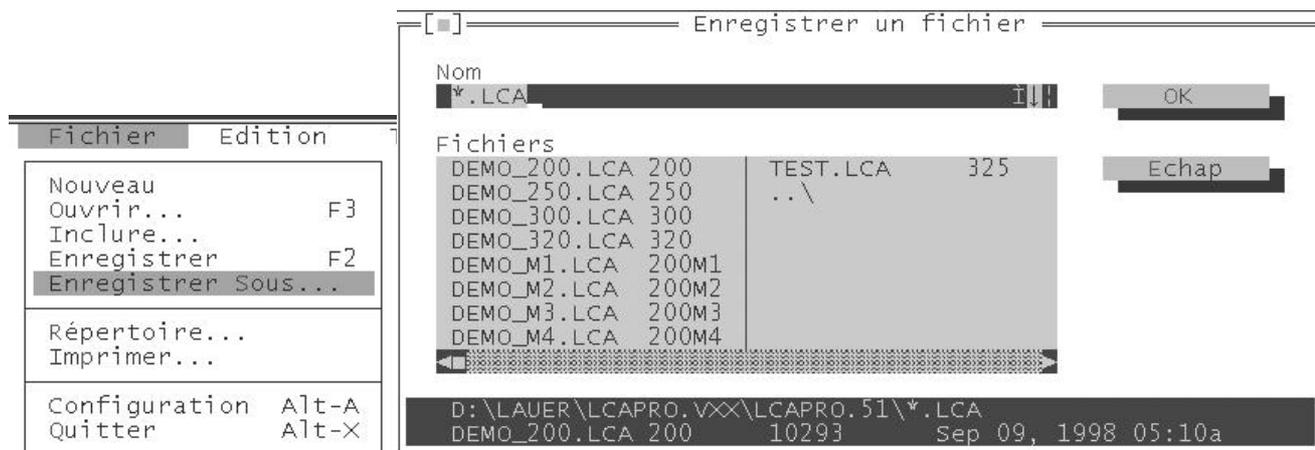
Le temps de modification des variables permet de sortir des textes Menus si aucune touche du terminal est actionnée.

Si cette valeur est égale à 0, la sortie automatique des textes menus est désactivée.

9.9. Sauvegarde et transfert de l'application

Une fois l'application terminée une sauvegarde est nécessaire avant le transfert.

Le Menu <<FICHIER>> et la commande "Enregistrer Sous" permet de sauvegarder l'application.



Le transfert de l'application vers le terminal LCA se réalise par le menu <<TRANSFERT>> et la commande "Transfert".

9.10. Assistance technique

N'hésitez pas à nous appeler si vous rencontrez une difficulté lors de la mise en œuvre de notre matériel ; nous sommes en permanence à votre service :

Le logiciel donne grâce au menu <<INFO>> et la commande "A propos" les coordonnées de la société IVO-industries.

Christian HECHT
Ingénieur Produits

Francis WERCK
Ingénieur Développement



Adresse : E-mail : techni@ivo-industries.fr

10. CONFIGURATION AUTOMATE

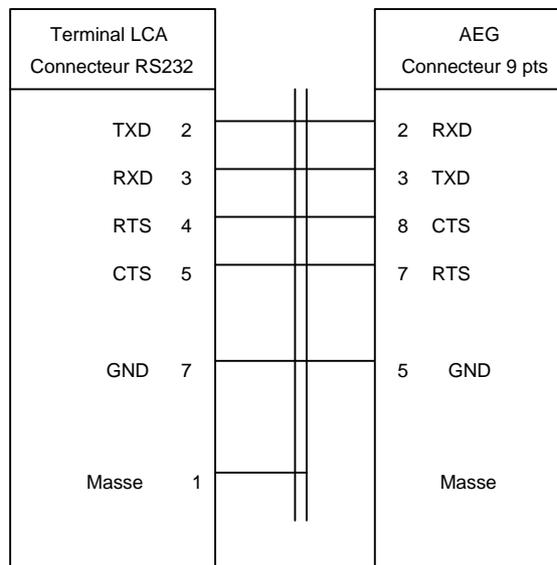
10.1. Automate AEG

10.1.1. Protocole Bkos

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate AEG A120, A250.

Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 778 pour automate AEG A120, A250

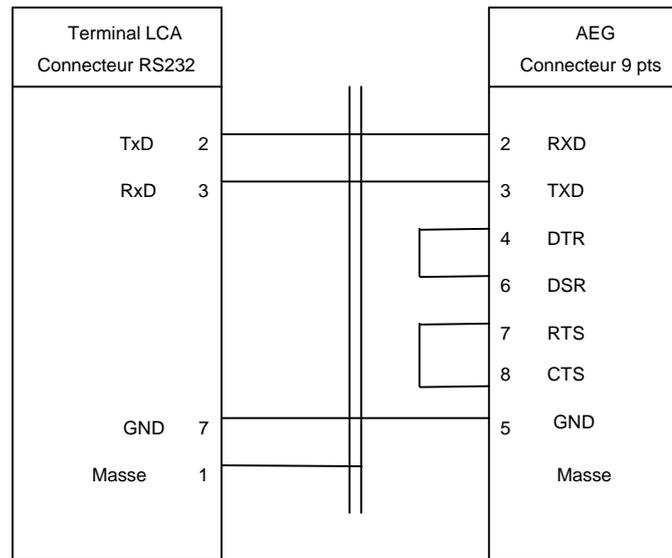


Les paramètres de la liaison série, pour un automate AEG, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET-PARAMETRES PROTOCOLE**'.

10.1.2. Protocole Modbus-1

A utiliser dès que le terminal LCA est connecté en Modbus sur un automate AEG.
 Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 780 pour automate AEG Modbus



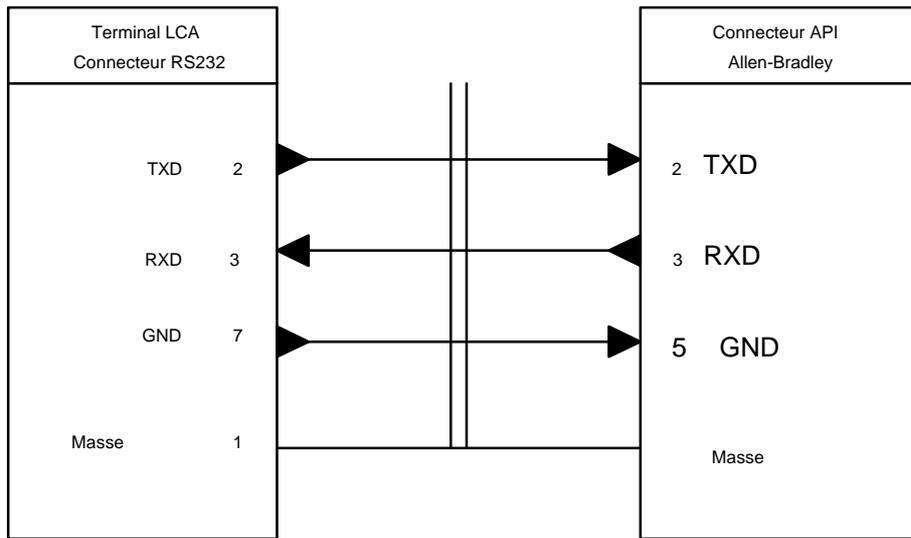
Les paramètres de la liaison série, pour un automate AEG, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET-PARAMETRES PROTOCOLE**'.

10.2. Automate ALLEN-BRADLEY

Protocole DH 485

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Allen-Bradley. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 709 pour automate ALLEN-BRADLEY SLC 5/02...5/04



Les paramètres de la liaison série, pour un automate Allen-Bradley, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

La vitesse de communication est de 19200 bauds.

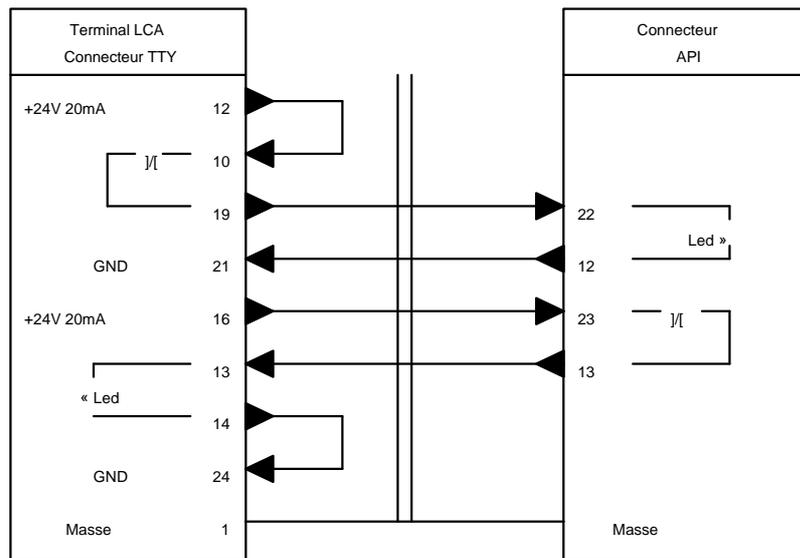
Pour les automates SLC 504 canal 0 sélectionnez dans l'automate le protocole "**PIC-1**"

10.3. Automate BOSCH

Protocole BUEP19E

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate BOSCH. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 706 pour automate BOSCH



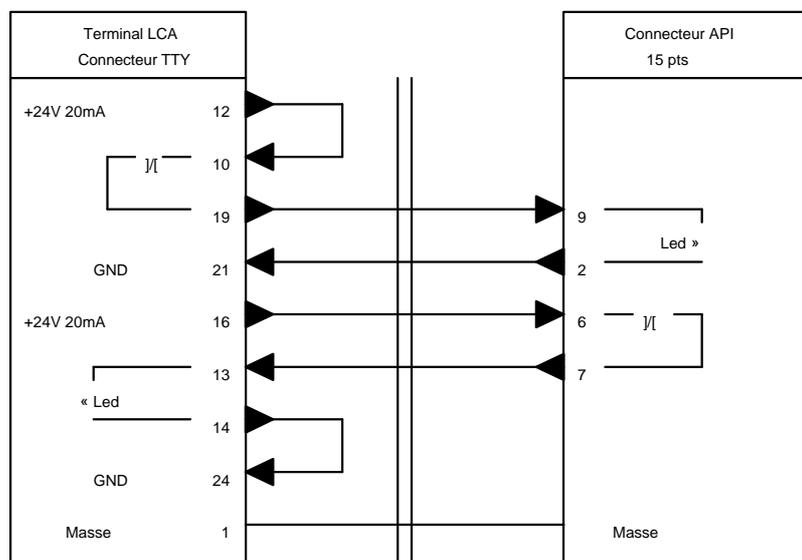
Les paramètres de la liaison série, pour un automate BOSCH, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

10.4. Automate FESTO

10.4.1. Protocole AS511

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Festo ISB50. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 716 pour automate FESTO ISB 50



Les paramètres de la liaison série, pour un automate Festo IBS 50, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

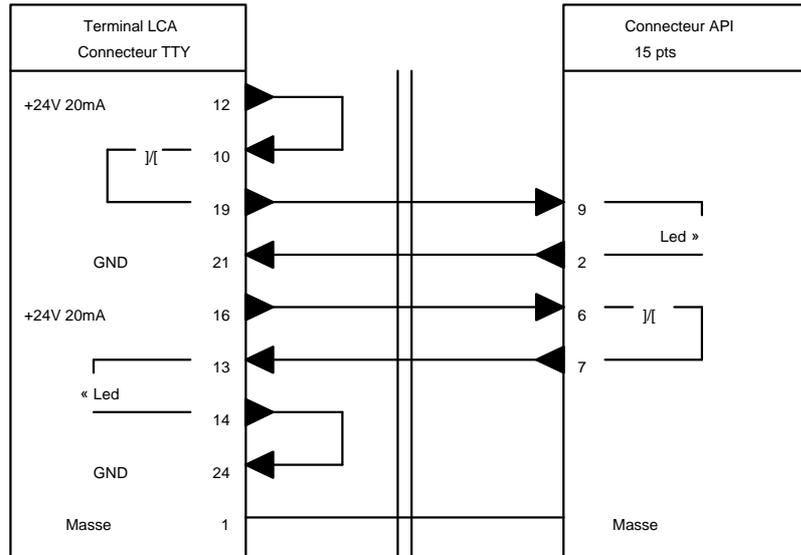
10.4.2. Protocole L1

A utiliser dès que le terminal LCA est connecté sur une prise de programmation d'un automate Festo IBS 50 en protocole L1.

Ce protocole nécessite un programme automate pour la communication entre le terminal LCA et l'automate.

Ce programme se trouve sur la disquette 'LCAPRO disk 2/2' et le fichier porte le nom LCAAPI.EXE Cette disquette est jointe avec le logiciel de programmation des terminaux LCA.

Câble de liaison PCS 716 pour automate FESTO IBS 50

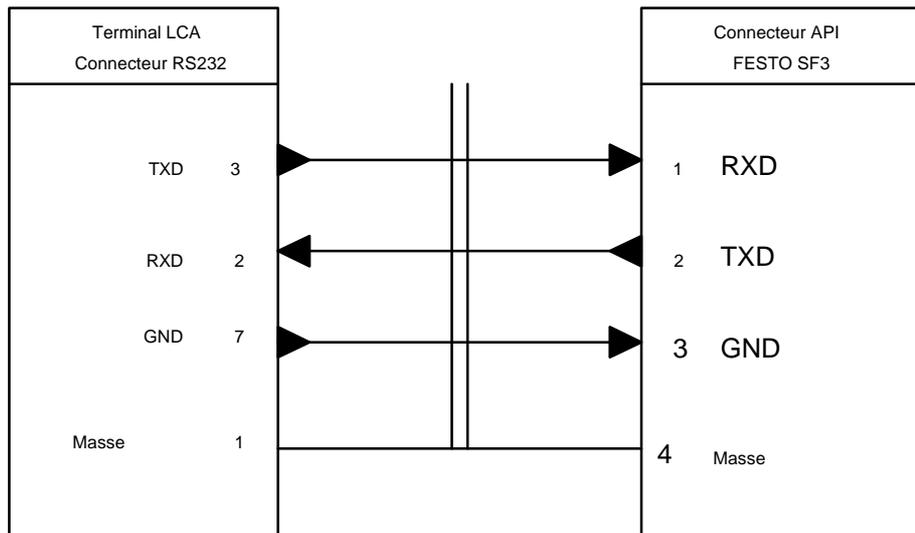


10.4.3 Protocole LAUER

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Festo. Ce protocole nécessite un programme automate pour la communication entre le terminal LCA et l'automate.

Ce programme se trouve sur la disquette 'LCAPRO disk 2/2' et le fichier porte le nom LCAAPI.EXE Cette disquette est jointe avec le logiciel de programmation des terminaux LCA.

Câble de liaison PCS 779 pour automate FESTO SF3



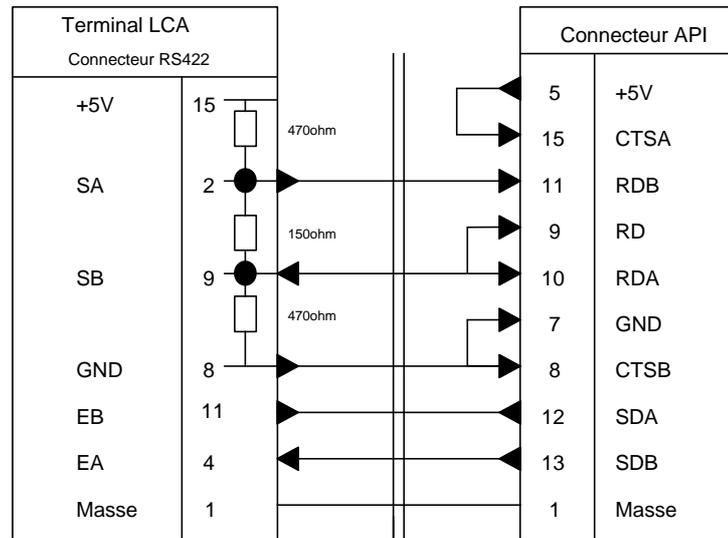
Les paramètres de la liaison série, pour un automate FESTO, se programment à l'aide du logiciel LCAPRO dans la fenêtre 'PROJET - PARAMETRES PROTOCOLE'.

10.5. Automate GE FANUC

Protocole SNP X

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate GE FANUC. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 708 pour automate GE FANUC



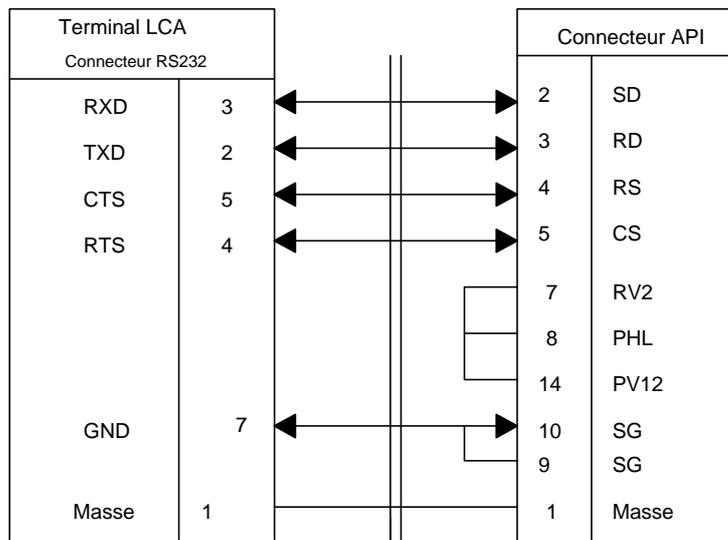
Les paramètres de la liaison série, pour un automate GE FANUC, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

10.6. Automate HITACHI

Protocole TCP1

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate HITACHI. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 722 pour automate HITACHI



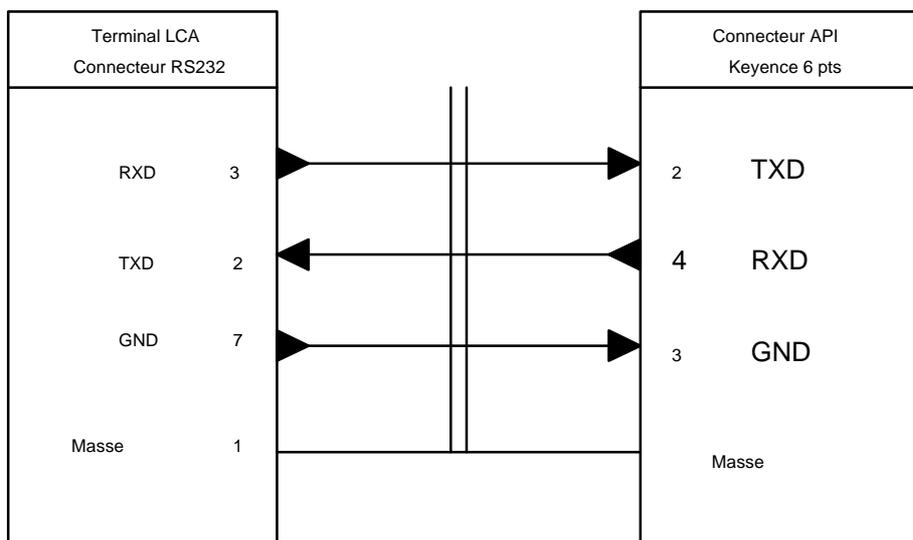
Les paramètres de la liaison série, pour un automate HITACHI, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

10.7. Automate KEYENCE

Protocole série KV

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Keyence. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 731 pour automate KEYENCE KV10...KV80



Les paramètres de la liaison série, pour un automate Keyence, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

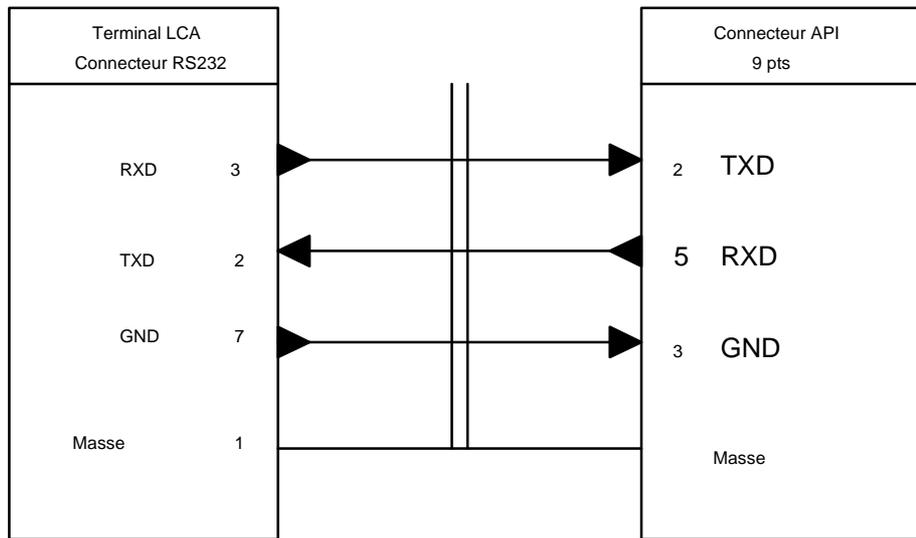
10.8. Automate KLÖCKNER- MOELLER

10.8.1. Protocole Sucom-A

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Klöckner-Moeller.

Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 785 pour automate KLÖCKNER-MOELLER PS4-201-MM1



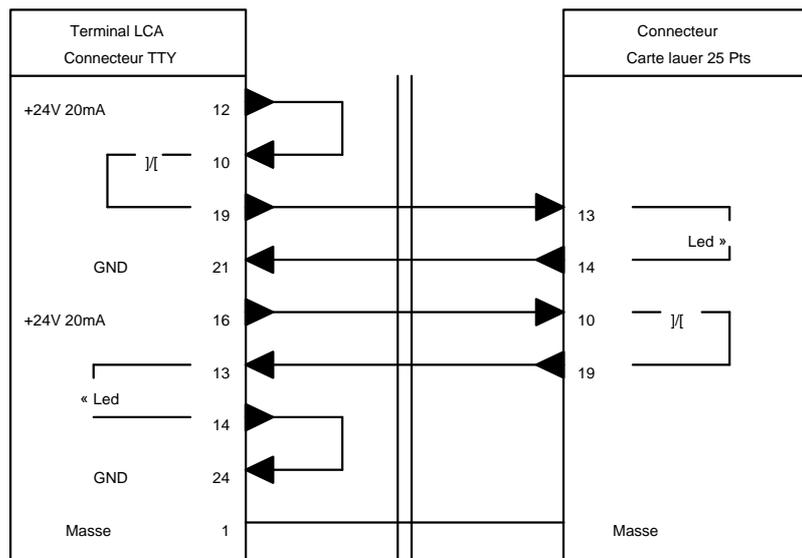
Les paramètres de la liaison série, pour un automate Klöckner-Moeller, se programment à l'aide du logiciel LCAPRO dans la fenêtre '**PROJET - PARAMETRES PROTOCOLE**'.

10.8.2. Protocole LAUER

A utiliser dès que le terminal LCA est connecté sur une prise de la carte de communication LAUER.
Ce protocole nécessite un programme automate pour la communication entre le terminal LCA et l'automate.

Ce programme se trouve sur la disquette 'LCAPRO disk 2/2' et le fichier porte le nom LCAAPI.EXE Cette disquette est jointe avec le logiciel de programmation des terminaux LCA.

Câble de liaison PCS 726 pour automate KLÖCKNER-MOELLER



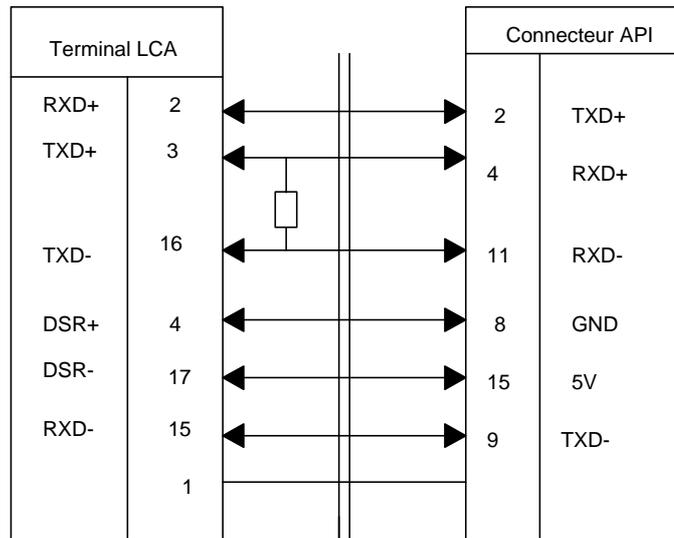
Les paramètres de la liaison série, pour un automate KLÖCKNER-MOELLER, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

10.9. Automate MITSUBISHI

Protocole FX

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate MITSUBISHI. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 748 pour automate MITSUBISHI



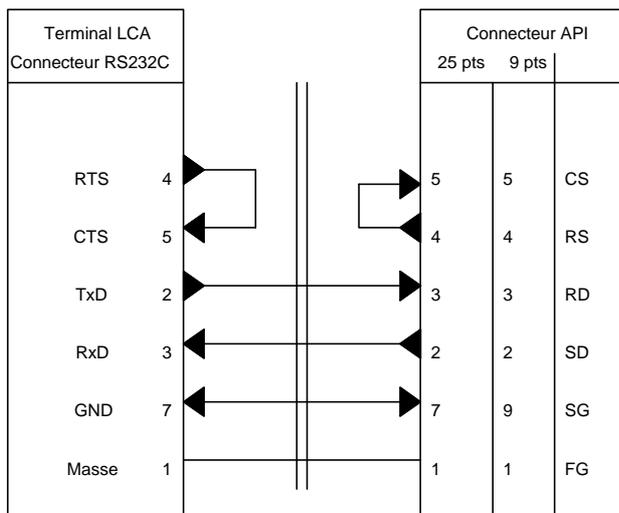
Les paramètres de la liaison série, pour un automate MITSUBISHI, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

10.10. Automate OMRON

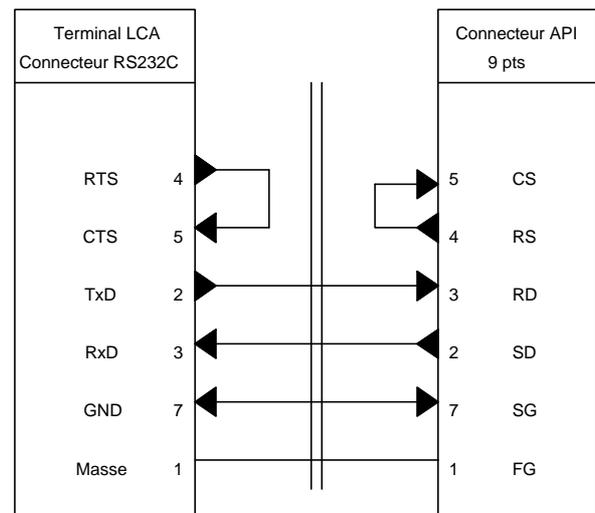
Protocole Host Link

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Omron. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 746 pour automate OMRON



Câble de liaison PCS 747 / automate OMRON CQM1



Les paramètres de la liaison série, pour un automate Omron, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE** '.

Configuration de l'automate

Il est nécessaire de configurer la liaison série des automates OMRON SYSMAC série C pour assurer la communication avec le terminal LCA.

* L'automate SYSMAC CQM1

Pour l'automate CQM1 la communication est réalisée par la liaison série de l'automate. La vitesse de transmission du terminal LCA est configuré par défaut à 19200 Bauds, il faut également programmer la vitesse et le format des données dans l'automate. Cette vitesse se configure dans la mémoire donnée DM6645 = 1 et DM6646 = 0304, DM6648 et DM6653 = 0. (19200 bauds, 1 bit de start, 7bits de données, 2 bits de stop, parité paire).

*** Les automates SYSMAC CXXH**

Pour les automates CXXH la communication est réalisée par la liaison série de l'automate. La vitesse de transmission du terminal LCA est configuré par défaut à 9600 Bauds, il faut également programmer la même vitesse de transmission dans l'automate. Cette vitesse se configure dans la mémoire donnée DM921: bits 0 à 7.

Valeur	Vitesse
2	1200 Bauds
4	4800 Bauds
5	9600 Bauds

*** L'automate SYSMAC C200H**

Pour cet automate la communication est réalisée à l'aide de la carte C200H-LK201 à mettre dans l'automate. Cette carte doit être configurée de la façon suivante :

Configuration en face arrière	Configuration en face avant
DIP-Switch : 1 à 4 sur ON	SW1, 2, 4 : sur 0
CTS-Sélecteur : sur 0	SW3 : identique à la vitesse du terminal LCA Position 2 1200 Bauds Position 4 4800 Bauds Position 5 9600 Bauds Position 6 19200 Bauds

***Les automates SYSMAC C1000H, C2000H, C2000**

Pour ces automates la communication est réalisée à l'aide de la carte **3G2A5-LK201-EV1** à mettre dans l'automate. Cette carte doit être configurée de la façon suivante :

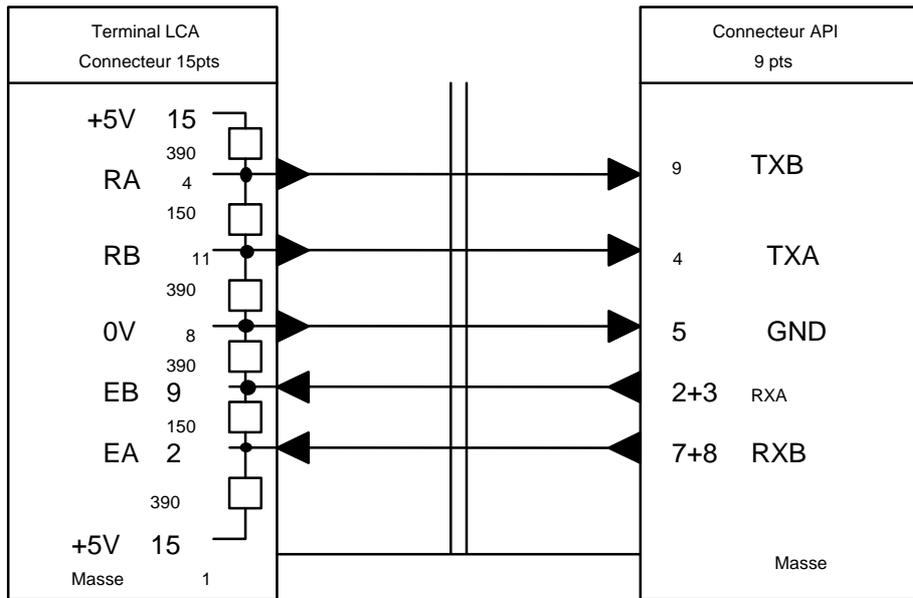
Configuration	
Désignation	Position
I/O port sélecteur	RS232C
Synchronisation	Internal
Switch de la résistance de terminaison	OFF
CTS-Sélecteur	0V
SW1	DIL 1...7 sur OFF DIL 8 sur ON
SW2	DIL 5...8 sur OFF DIL 1, 2, 3, 4 vitesse de transmission 0 0 0 1 1200 Bauds 0 1 1 0 2400 Bauds 1 0 1 0 9600 Bauds 0 0 1 0 19200 Bauds (0 = OFF 1 = ON)

10.11. Automate PHILIPS

Protocole PPCCOM

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Philips.
Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 795 pour automate PHILIPS P8-MMS102



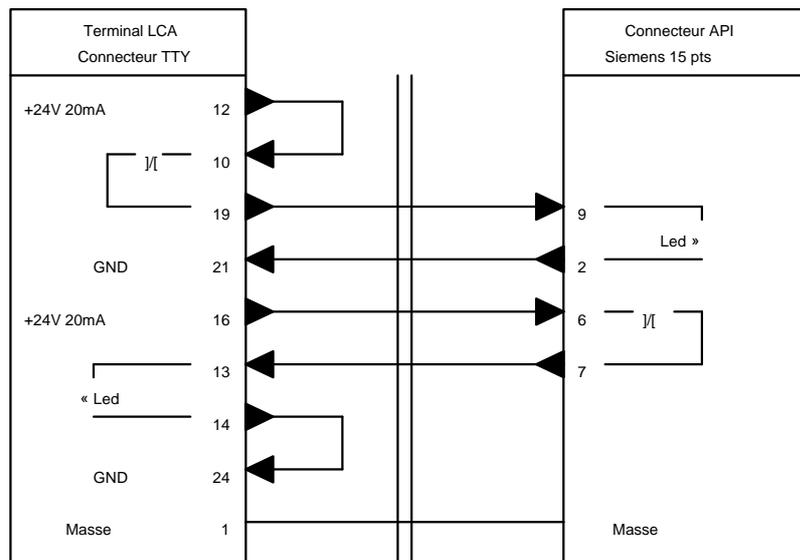
Les paramètres de la liaison série, pour un automate Philips, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE**'.

10.12. Automate SIEMENS S5

10.12.1. Protocole AS511

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Siemens S5. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 716 pour automate SIEMENS S5



Les paramètres de la liaison série, pour un automate de la série S5, se programment à l'aide du logiciel LCAPRO dans la fenêtre 'PROJET - PARAMETRES PROTOCOLE'.

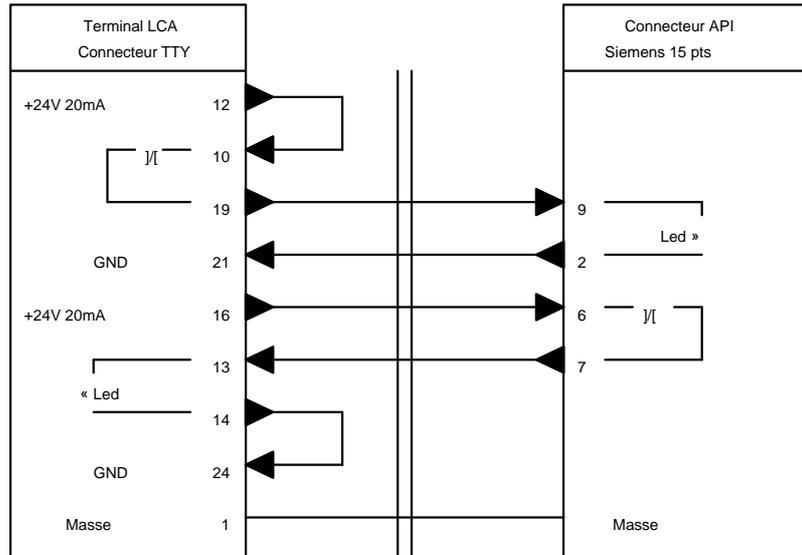
10.12.2. Protocole L1

A utiliser dès que le terminal LCA est connecté sur une prise de programmation d'un automate S5 en protocole L1.

Ce protocole nécessite un programme automate pour la communication entre le terminal LCA et l'automate.

Ce programme se trouve sur la disquette 'LCAPRO disk 2/2' et le fichier porte le nom LCAAPI.EXE Cette disquette est jointe avec le logiciel de programmation des terminaux LCA.

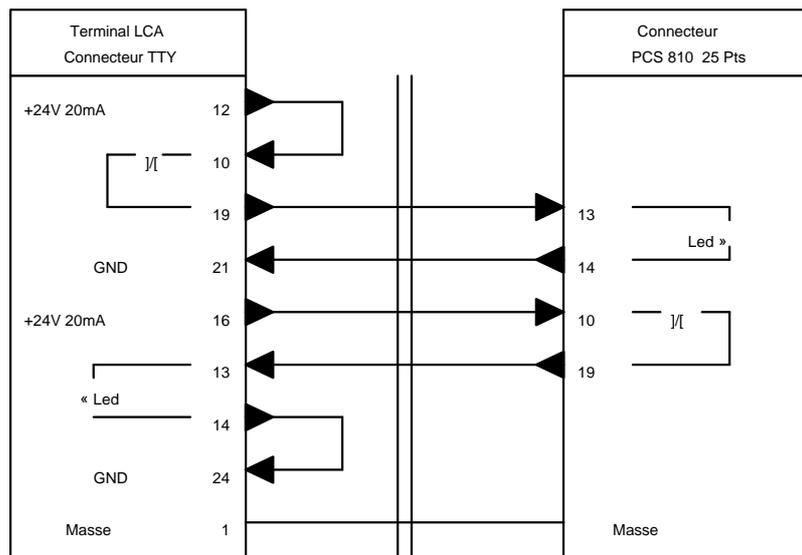
Câble de liaison PCS 716 pour automate SIEMENS S5



10.12.3. Protocole LAUER

A utiliser dès que le terminal LCA est connecté sur une prise de la carte de communication PCS 810. Ce protocole nécessite un programme automate pour la communication entre le terminal LCA et l'automate. Ce programme se trouve sur la disquette 'LCAPRO disk 2/2' et le fichier porte le nom LCAAPI.EXE Cette disquette est jointe avec le logiciel de programmation des terminaux LCA.

Câble de liaison PCS 736 pour automate SIEMENS S5



Les paramètres de la liaison série, pour un automate de la série S5, se programment à l'aide du logiciel LCAPRO dans la fenêtre 'PROJET - PARAMETRES PROTOCOLE'.

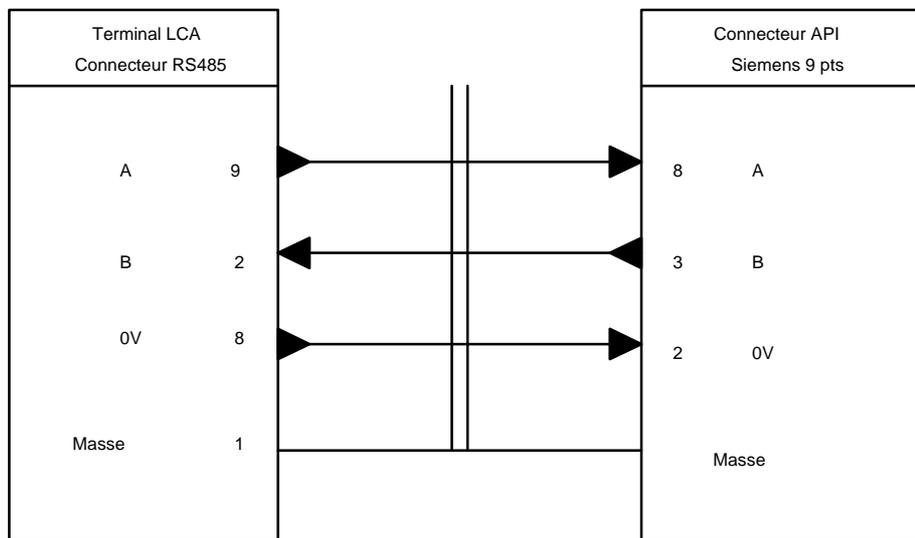
10.13. Automate SIEMENS S7

Protocole S7-200

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Siemens S7-200.

Ce protocole nécessite un programme automate pour la communication entre le terminal LCA et l'automate. Ce programme se trouve sur la disquette 'LCAPRO disk 2/2' et le fichier porte le nom LCAAPI.EXE Cette disquette est jointe avec le logiciel de programmation des terminaux LCA.

Câble de liaison PCS 721 pour automate SIEMENS S7-200



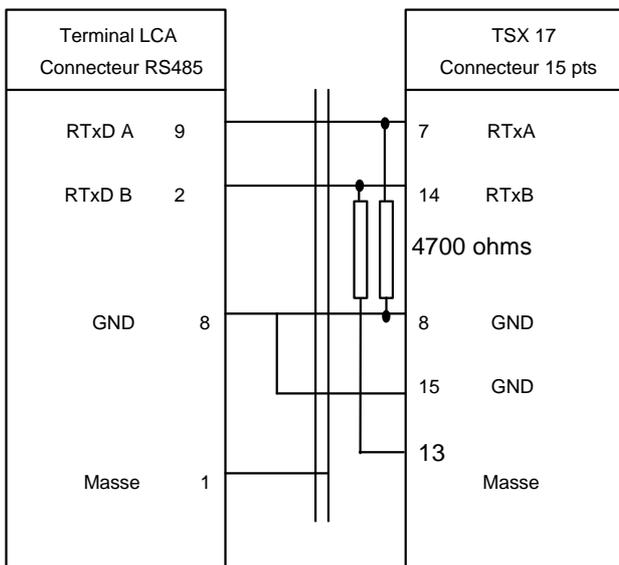
Les paramètres de la liaison série, pour un automate de la série S7-200, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE**'.

10.14. Automate TELEMECANIQUE

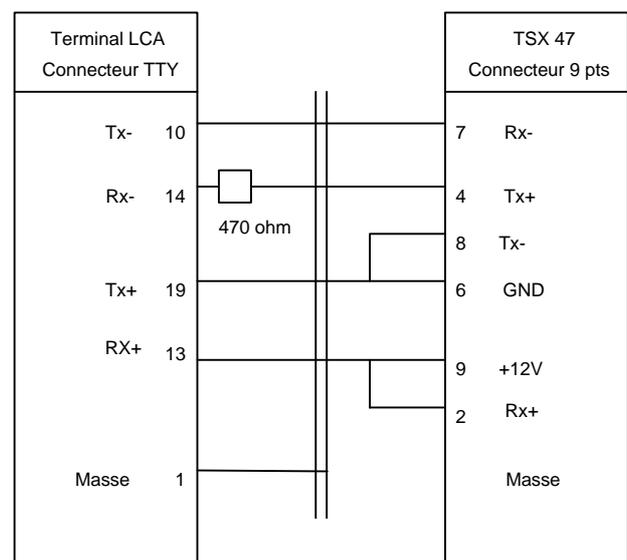
10.14.1. Protocole TSX-direct

A utiliser dès que le terminal LCA est connecté sur la prise de programmation d'un automate Télémécanique. Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 758 pour automate Télémécanique TSX 17-20



Câble de liaison PCS 759 pour automate Télémécanique TSX

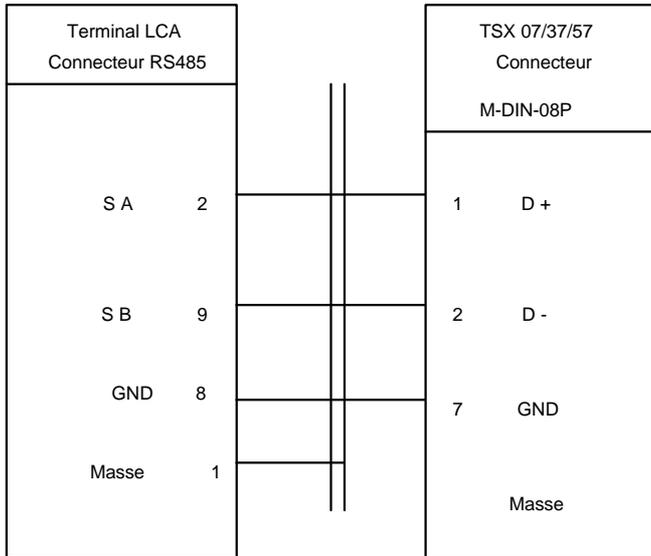


Les paramètres de la liaison série, pour un automate Télémécanique, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET PARAMETRES PROTOCOLE** '.

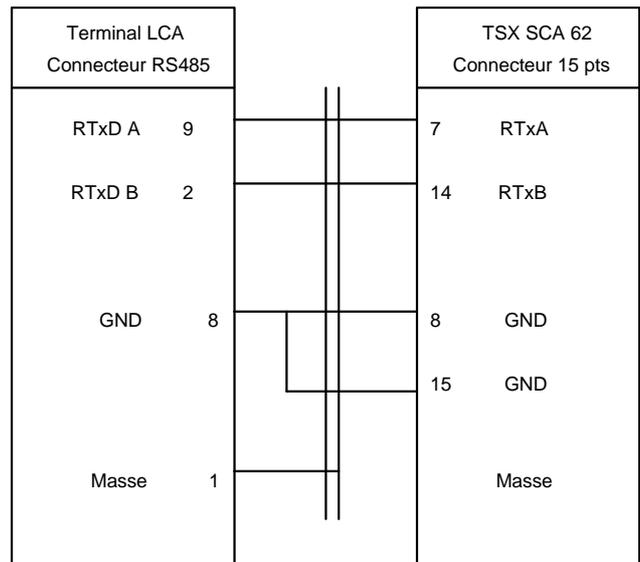
10.14.2. Protocole Unitelway

A utiliser dès que le terminal LCA est connecté sur la prise Unitelway d'un automate Télémécanique.
Ce protocole ne nécessite aucun programme automate pour la communication entre le terminal LCA et l'automate.

Câble de liaison PCS 783 pour automate Télémécanique TSX 07/37/57



Câble de liaison PCS 718 pour automate Télémécanique



Les paramètres de la liaison série, pour un automate Télémécanique, se programment à l'aide du logiciel LCAPRO dans la fenêtre ' **PROJET - PARAMETRES PROTOCOLE**'.

Les paramètres sont les suivant :

- Numéro **esclave** du terminal " Numéro de station LCA :"
- Numéro de l'automate " Numéro de station API"
- Vitesse de communication " Baudrate 9600 " pour TSX 37
19200" pour TSX 57

Nota 1 :

- Pour un automate TSX17-20, TSX 07, TSX 37 vitesse maximum autorisée 9600 bauds.

Nota 2 :

Afin de pouvoir communiquer simultanément, sur l'automate, avec le terminal et le PC, il faut donner, au terminal, un numéro d'esclave supérieur à 3. Le PC utilisant les adresses 1,2,3 lors du transfert d'un programme vers l'automate ou lors de la mise au point du programme.

