

Manuel de Référence

PL7Micro/Junior/Pro

Annexes

TLX DR PL7 xx fre

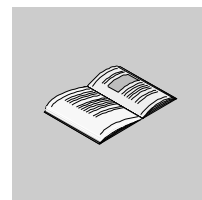
Documents à consulter

Présentation

Ce manuel se compose de trois tomes:

- Tome 1: Description du logiciel PL7
 - Généralités
 - Langage à contacts
 - Langage liste d'instructions
 - Langage littéral structuré
 - Langage Grafcet
 - Blocs fonction DFB
 - Modules Fonctionnels
 - Tome 2: Description détaillée des instructions et des fonctions
 - Instructions de base
 - Instructions avancées
 - Objets bits et mots système
 - Tome 3: Annexes
 - Différences entre PL7-2/3 et PL7-Micro/Junior
 - Aide-mémoire
 - Liste des mots réservés
 - Conformité au standard CEI 1131-3
 - Serveur OLE Automation
 - Performances
-

Table des matières



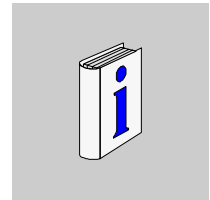
| | | |
|-------------------|--|-----------|
| | A propos de ce manuel | 9 |
| Chapitre 1 | Différences entre PL7-2/3 et PL7 Micro/Junior | 11 |
| | Présentation | 11 |
| | Valeurs immédiates et étiquettes | 12 |
| | Bits | 13 |
| | Mots | 15 |
| | Blocs fonction | 17 |
| | Tableaux de bits et de mots | 20 |
| | Blocs fonctions optionnels | 21 |
| | Instructions | 22 |
| | Délimiteurs | 26 |
| Chapitre 2 | Aide-mémoire | 27 |
| | Présentation | 27 |
| | Instructions booléennes | 28 |
| | Instructions ST | 30 |
| | Blocs fonctions LD et IL | 31 |
| | Blocs fonctions ST | 32 |
| | Structures de contrôle ST | 33 |
| | Arithmétique entière (simple et double longueur) | 34 |
| | Arithmétique sur flottants | 35 |
| | Conversions numériques | 36 |
| | Tableaux de bits | 37 |
| | Instructions sur tableaux | 38 |
| | Instructions sur tableaux de flottants | 39 |
| | Instructions "Orphée" | 40 |
| | Echanges explicites | 41 |
| | Instructions de gestion du temps | 42 |
| | Instructions de temporisation | 43 |
| | Instructions d'archivage de données | 44 |
| | Instructions sur chaînes de caractères | 45 |
| | Multi-tâches et événements | 46 |
| | Communication | 47 |

| | | |
|-------------------|---|-----------|
| Chapitre 3 | Mots réservés | 49 |
| | Mots réservés | 49 |
| Chapitre 4 | Conformité au standard CEI1131-3 | 55 |
| | Présentation | 55 |
| | Conformité au standard CEI 1131-3 | 56 |
| | Eléments commun | 57 |
| | Eléments de langage IL | 63 |
| | Eléments de langage ST | 64 |
| | Eléments graphiques communs | 65 |
| | Eléments de langage LD | 66 |
| | Paramètres dépendant de l'implémentation | 67 |
| | Situations d'erreurs | 70 |
| Chapitre 5 | Serveur OLE Automation | 71 |
| | Présentation | 71 |
| 5.1 | Présentation | 72 |
| | Présentation | 72 |
| | Présentation du Serveur OLE Automation | 73 |
| | Modes de marche du serveur OLE Automation | 75 |
| 5.2 | Mise en oeuvre | 76 |
| | Présentation | 76 |
| | Installation de OLE Automation | 77 |
| | Mécanisme d'accès au serveur PL7 OLE Automation | 78 |
| | Lancement du serveur en mode local (mode COM) | 79 |
| | Lancement du serveur en mode distant (DCOM) | 80 |
| | Mise en oeuvre du serveur en mode distant | 81 |
| | Modes d'exécution PL7 serveur | 82 |
| | Points d'entrée : Fonction OLE | 83 |
| 5.3 | Fonctions OLE | 85 |
| | Présentation | 85 |
| | OpenStx | 87 |
| | CloseStx | 88 |
| | OpenStxWithMaj | 89 |
| | ExportScyFile | 90 |
| | ExportFeFile | 91 |
| | ExportFefFileWithConfiguration | 92 |
| | DisconnectPLC | 93 |
| | ConnectPLC | 94 |
| | SaveStx | 95 |
| | DownloadDataToPLC | 96 |
| | DownloadToPLC | 98 |
| | UploadFromPLC | 99 |
| | UploadDataFromPLC | 100 |
| | UploadFromPLCWithSymbol | 102 |
| | GetSymbol | 104 |

| | | |
|-------------------|---|------------|
| | SetServerIHM | 105 |
| | SetDownloadFunction | 106 |
| | GetPL7State | 107 |
| | GetSTXApplIdentity | 108 |
| | GetPLCApplIdentity | 109 |
| | SendCommandToPLC | 111 |
| | SetDriverAndAdresse | 112 |
| | OpenTool | 113 |
| | SetPosPL7Windows | 115 |
| | ShowProgram | 116 |
| | CloseProgram | 117 |
| | ShowIOModule | 118 |
| | CloseIOModule | 119 |
| | ShowDFB | 120 |
| | CloseDFB | 121 |
| | GetMessageError | 122 |
| | GetServerVersion | 123 |
| | ImportScyFile | 124 |
| Chapitre 6 | Performances | 125 |
| | Présentation | 125 |
| 6.1 | Informations générales | 126 |
| | Principes de calcul | 126 |
| 6.2 | Performances des instructions sur automates Micro | 128 |
| | Présentation | 128 |
| | Performances des instructions booléennes | 129 |
| | Performances des blocs fonction | 131 |
| | Arithmétique entière et à virgule flottante | 134 |
| | Instructions sur programme et structures de contrôle | 138 |
| | Conversions numériques | 140 |
| | Instructions sur chaîne de bits | 141 |
| | Instruction sur tableaux de mots, doubles mots et flottants | 144 |
| | Instructions de gestion du temps | 151 |
| | Instructions sur chaînes de caractères | 153 |
| | Fonctions métier et fonctions Orphée | 155 |
| | Instructions d'entrées/sorties explicite | 158 |
| 6.3 | performances des instructions sur automates Premium | 159 |
| | Présentation | 159 |
| | Performances des instructions booléennes | 160 |
| | Performances des blocs fonction | 164 |
| | Arithmétique entière et à virgule flottante | 167 |
| | Instructions sur programme et structures de contrôle | 171 |
| | Conversions numériques | 173 |
| | Instructions sur chaîne de bits | 174 |
| | Instruction sur tableaux de mots, doubles mots et flottants | 177 |
| | Instructions de gestion du temps | 185 |

| | | |
|--------------|---|------------|
| | Instructions sur chaînes de caractères | 187 |
| | Fonctions métier et fonctions Orphée | 190 |
| | Instructions d'entrées/sorties explicite | 193 |
| | Bloc fonction DFB | 195 |
| 6.4 | Fonctions avancées | 198 |
| | Présentation | 198 |
| | Description des zones mémoire | 199 |
| | Taille mémoire des objets | 200 |
| | Bilan des consommations mémoire des modules sur Micro | 201 |
| | Bilan des consommations mémoire des modules sur Premium | 204 |
| | Taille mémoire des fonctions avancées | 210 |
| | Méthode de calcul du nombre d'instructions | 220 |
| Index | | 229 |

A propos de ce manuel



Présentation

Objectif du document

Ce manuel donne un ensemble d'informations complémentaires à la programmation des automates Micro, Premium et Atrium.

Champ d'application

La mise à jour de cette publication prend en compte les fonctionnalités de PL7 V4.3; Elle permet néanmoins de mettre en oeuvre les versions antérieures de PL7.

Commentaires utilisateur

Envoyez vos commentaires à l'adresse e-mail TECHCOMM@modicon.com

Différences entre PL7-2/3 et PL7 Micro/Junior

1

Présentation

Contenu de ce chapitre

Ce chapitre décrit les différences des objets et instructions entre les logiciels PL7-2/PL7-3 et PL7 Micro/PL7 Junior

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

| Sujet | Page |
|----------------------------------|------|
| Valeurs immédiates et étiquettes | 12 |
| Bits | 13 |
| Mots | 15 |
| Blocs fonction | 17 |
| Tableaux de bits et de mots | 20 |
| Blocs fonctions optionnels | 21 |
| Instructions | 22 |
| Délimiteurs | 26 |

Valeurs immédiates et étiquettes

Valeurs immédiates

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|----------------------|------------------|------------------|
| Entier base 10 | 1234 | 1234 |
| Entier base 2 | L'10011110' | 2#10011110 |
| Entier base 16 | H'ABCD' | 16#ABCD |
| Flottant | -1.32e12 (PL7-3) | -1.32e12 |
| Chaîne de caractères | M'aAbBcC' | `aAbBcC' |

Étiquettes

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|--------|----------------|------------------|
| Label | Li i = 0 à 999 | %Li i = 0 à 999 |

Bits

Bits d'Entrée en bac

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|--------------------------------|------------------------|--------------------|
| Bit d'entrée en bac | Ixy, i | %Ixy.i |
| Bit d'entrée en bac indexé | Ixy, i (Wj) (PL7-3) | %Ixy.i[%MWj] |
| Bit d'entrée à distance | RIx, y, i (PL7-3) | %I\chemin\mod.voie |
| Bit d'entrée à distance indexé | RIx, y, i (Wj) (PL7-3) | - |

Bits de Sortie en bac

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|---------------------------------|------------------------|--------------------|
| Bit de sortie en bac | Oxy, i | %Qxy.i |
| Bit de sortie en bac indexé | Oxy, i (Wj) (PL7-3) | %Qxy.i[%MWj] |
| Bit de sortie à distance | ROx, y, i (PL7-3) | %Q\chemin\mod.voie |
| Bit de sortie à distance indexé | ROx, y, i (Wj) (PL7-3) | - |

Bits défaut d'E/S en bac

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|-------------------|-----------------|------------------|
| Bit défaut module | Ixy, S / Oxy, S | %Ixy.MOD.ERR |
| Bit de voie | - | %Ixy.i.ERR |

Bits défaut d'E/S à distance

Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 (uniquement) | PL7 Micro/Junior |
|--------------------------------|--------------------------|------------------------|
| Bit défaut module | - | %I\chemin\mod.MOD.ERR |
| Bit de voie | RDX, y, i / ERRORx, y, i | %I\chemin\mod.voie.ERR |
| bit disjonction voie de sortie | TRIPx, y, i | - |
| bit réarmement voie de sortie | RSTx, y, i | - |

Bits internes et bits système Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|--------------------|-----------------|------------------|
| Bit interne | Bi | %Mi |
| Bit interne indexé | Bi (Wj) (PL7-3) | %Mi [%MWj] |
| Bit système | SYi | %Si |

Bits d'étapes Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|--|---------------|------------------|
| Bit d'étape | Xi | %Xi |
| Bit de macro-étape | XMj (PL7-3) | %XMj |
| Bit d'étape i de macro-étape j | Xj, i (PL7-3) | %Xj. i |
| Bit d'étape d'entrée de macro-étape j | Xj, I (PL7-3) | %Xj. IN |
| Bit d'étape de sortie de macro-étape j | Xj, O (PL7-3) | %Xj. OUT |

Bits de mots Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|---------------------------------------|---|-------------------------------|
| Bit j du mot interne i | Wi, j | %MWi : Xj |
| Bit j du mot interne i indexé | Wi (Wk), j (PL7-3) | %MWi [%MWk] : Xj |
| Bit j du mot constant i | CWi, j | %KWi : Xj |
| Bit j du mot constant i indexé | CWi (Wk), j (PL7-3) | %KWi [%MWk] : Xj |
| Bit j du registre i | IW/OWxy, i, j | %IW/%QWxy. i : Xj |
| Bit k du mot commun j de la station i | COMi, j, k COMXi, j, k (X = B, C, D) | %NWi. j : Xk %NXWi. j : Xk |
| Bit j du mot système i | SWi, j | %SWi : Xj |

Mots

Mots internes Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|------------------------------------|-----------------|------------------|
| Mot interne simple longueur | wi | %MWi |
| Mot interne simple longueur indexé | wi(wj) (PL7-3) | %MWi [%MWj] |
| Mot interne double longueur | DWi (PL7-3) | %MDi |
| Mot interne double longueur indexé | DWi(wj) (PL7-3) | %MDi [%MWj] |
| Mot interne réel | - | %MFi |
| Mot interne réel indexé | - | %MFi [%MWj] |

Mots constants Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|-------------------------------------|------------------|------------------|
| Mot constant simple longueur | CWi | %KWi |
| Mot constant simple longueur indexé | CWi(wj) | %KWi [%MWj] |
| Mot constant double longueur | CDWi (PL7-3) | %KDi |
| Mot constant double longueur indexé | CDWi(wj) (PL7-3) | %KDi [%MWj] |
| Mot constant réel | - | %KFi |
| Mot constant réel indexé | - | %KFi [%MWj] |

Mots registre Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|--|--------------------|---------------------|
| Mot registre d'entrée simple longueur | IWxy, i | %IWxy. i |
| Mot registre d'entrée double longueur | - | %IDxy. i |
| Mot registre de sortie simple longueur | OWxy, i | %QWxy. i |
| Mot registre de sortie double longueur | - | %QDxy. i |
| Mot registre d'une entrée à distance | RIWx, y, i (PL7-3) | %IW\chemin\mod.voie |
| Mot registre d'une sortie à distance | ROWx, y, i (PL7-3) | %QW\chemin\mod.voie |

Autres mots Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 (uniquement) | PL7 Micro/Junior |
|---|--|-------------------------------------|
| Mot système | SWi | %SWi |
| Mot commun j de la station | COMi, j COMXi, j (avec X=B,C,D) | %NW{i}j %NW{[r.]i}j r=num de réseau |
| Mot d'état d'un module TOR distant | STATUSAx, y, i (PL7-3) STATUSBx, y, i (PL7-3) | - |
| Mot d'état d'une voie de module TOR distant | STSx, y, i (PL7-3) | %IW\chemin\mod.voie.ERR |

Blocs fonction

Temporisateur

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|------------------------------|---------|------------------|
| Repère | Ti | %Ti |
| Valeur de présélection (mot) | Ti , P | %Ti . P |
| Valeur courante (mot) | Ti , V | %Ti . V |
| Temporisateur en cours (bit) | Ti , R | %Ti . R |
| Temporisateur écoulé (bit) | Ti , D | %Ti . D |

Monostable

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|------------------------------|---------|------------------|
| Repère | Mi | %MNi |
| Valeur de présélection (mot) | Mi , P | %MNi . P |
| Valeur courante (mot) | Mi , V | %MNi . V |
| Monostable en cours (bit) | Mi , R | %MNi . R |

Compteur/ Décompteur

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|------------------------------|---------|------------------|
| Repère | Ci | %Ci |
| Valeur de présélection (mot) | Ci , P | %Ci . P |
| Valeur courante (mot) | Ci , V | %Ci . V |
| Débordement comptage (bit) | Ci , E | %Ci . E |
| Présélection atteinte (bit) | Ci , D | %Ci . D |
| Débordement décomptage (bit) | Ci , F | %Ci . F |

Registre Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|----------------------|---------|------------------|
| Repère | Ri | %Ri |
| Mot d'entrée (mot) | Ri , I | %Ri . I |
| Mot de sortie (mot) | Ri , O | %Ri . O |
| Registre plein (bit) | Ri , F | %Ri . F |
| Registre vide (bit) | Ri , E | %Ri . E |

Bloc Texte Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|--------|---------|--------------------|
| Repère | TXTi | Pas de blocs texte |

Programmateur cyclique Tableau des différences entre PL7-2 et PL7-Micro/Junior

| Objets | PL7-2 | PL7 Micro/Junior |
|---------------------------------------|------------|------------------|
| Repère | Di (PL7-2) | %DRi |
| Numéro du pas actif (mot) | Di , S | %DRi . S |
| temps d'activité du pas courant (mot) | Di , V | %DRi . V |
| 16 bits d'ordres (mot) | Di , Wj | %DRi . Wj |
| Dernier pas en cours (bit) | Di , F | %DRi . F |

**Compteur/
Temporisateur rapide** Tableau des différences entre PL7-2 et PL7-Micro/Junior

| Objets | PL7-2 | PL7 Micro/Junior |
|------------------------------|------------|------------------|
| Repère | FC (PL7-2) | - |
| Valeur de présélection (mot) | FC , P | - |
| Valeur courante (mot) | FC , V | - |
| Remise à 0 externe (bit) | FC , E | - |
| Présélection atteinte (bit) | FC , D | - |
| Comptage en cours (bit) | FC , F | - |

Horodateur Tableau des différences entre PL7-2 et PL7-Micro/Junior

| Objets | PL7-2 | PL7 Micro/Junior |
|---|--------------|-------------------------|
| Repère | H (PL7-2) | - |
| Type "WEEK" ou "YEAR" sélection jours LMMJVSD (mot) | VD | - |
| Consigne de début (mot) | BGN | - |
| Consigne de fin (mot) | END | - |
| Valeur courante < consigne (bit) | < | - |
| Valeur courante = consigne (bit) | = | - |
| Valeur courante > consigne (bit) | > | - |

Tableaux de bits et de mots

Chaînes de bits Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|--------------------------------|--------------------|------------------|
| Chaîne de bits internes | Bi [L] | %Mi :L |
| Chaîne de bits d'entrée | Ixy, i [L] (PL7-3) | %Ixy. i :L |
| Chaîne de bits de sortie | Oxy. i [L] (PL7-3) | %Qxy. i :L |
| Chaîne de bits d'étape Grafcet | Xi [L] (PL7-3) | %Xi :L |
| Chaîne de bits de macro-étape | XMi [L] (PL7-3) | - |

Chaînes de caractères Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|--------|---------|-----------------------|
| Chaîne | - | %MBi :L (avec i pair) |

Tableaux de mots Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|---|----------------------------|------------------|
| Tableau de mots internes | Wi [L] | %MWi :L |
| Tableau de mots internes indexés | Wi (Wj) [L] | %MWi [%MWj] :L |
| Tableau de doubles mots internes | DWi [L] (PL7-3) | %MDi :L |
| Tableau de doubles mots internes indexés | DWi (Wj) [L] (PL7-3) | %MDi [%MWj] :L |
| Tableau de mots constants | CWi [L] | %KWi :L |
| Tableau de mots constants indexés | CWi (Wj) [L] | %KWi [%MWj] :L |
| Tableau de doubles mots constants | CDWi [L] (PL7-3) | %KDi :L |
| tableau de doubles mots constants indexés | CDWi (Wj) [L] (PL7-3) | %KDi [%MWj] :L |
| Tableau de réels | - | %MFi :L |
| Tableau de réels indexés | - | %MFi [%MWj] :L |
| Tableau de réels constants | - | %KFi :L |
| Tableau de réels constants indexés | - | %KFi [%MWj] :L |
| Tableau d'éléments d'entrée à distance | RIx, y, i [L] (PL7-3) | - |
| Tableau d'éléments de sortie à distance | ROx, y, i [L] (PL7-3) | - |
| Tableau d'éléments indexés d'entrée à distance | RIx, y, i (Wj) [L] (PL7-3) | - |
| Tableau d'éléments indexés de sortie à distance | ROx, y, i (Wj) [L] (PL7-3) | - |

Blocs fonctions optionnels

OFB

Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 (uniquement) | PL7 Micro/Junior |
|---------------------------------|----------------------------|------------------|
| Repère | < OFB >i | - |
| Elément d'OFB | < OFB >i, <élément> | - |
| Elément d'OFB indexé | < OFB >i, <élément>(Wj) | - |
| Tableau d'éléments d'OFB | < OFB >i, <élément>[L] | - |
| Tableau d'éléments d'OFB indexé | < OFB >i, <élément>(Wj)[L] | - |

Instructions

Instructions sur bits

Tableau des différences entre PL7-2, PL7-3 et PL7-Micro/Junior

| Objets | PL7-2 | PL7-3 | PL7 Micro/Junior |
|-------------------|-------|-------|------------------|
| Inversion logique | - | NOT | NOT |
| ET | AND | * | AND |
| OU | OR | + | OR |
| OU exclusif | XOR | - | XOR |
| Front montant | - | RE | RE |
| Front descendant | - | FE | FE |
| Mise à 1 | - | SET | SET |
| Mise à 0 | - | RESET | RESET |

Instructions sur mots et doubles mots

Tableau des différences entre PL7-2, PL7-3 et PL7-Micro/Junior

| Objets | PL7-2 | PL7-3 | PL7 Micro/Junior |
|------------------------------|---------------------|---------------------|---------------------|
| Addition | + | + | + |
| Soustraction | - | - | - |
| Multiplication | * | * | * |
| Division | / | / | / |
| Comparaison | >, >=, <, <=, =, <> | >, >=, <, <=, =, <> | >, >=, <, <=, =, <> |
| Reste d'une division | MOD | REM | REM |
| Racine carrée | - | SQRT | SQRT |
| Valeur absolue | - | - | ABS |
| ET logique | AND | AND | AND |
| OU logique | OR | OR | OR |
| OU logique exclusif | XOR | XOR | XOR |
| Complément logique | CPL | CPL | CPL |
| Incrémentation | - | INC | INC |
| Décrémentation | - | DEC | DEC |
| Décalage logique à gauche | - | SHL | SHL |
| Décalage logique à droite | - | SHR | SHR |
| Décalage circulaire à gauche | SLC | SLC | ROL |

| Objets | PL7-2 | PL7-3 | PL7 Micro/Junior |
|------------------------------|-------|-------|------------------|
| Décalage circulaire à droite | SRC | SRC | ROR |

Instructions de type flottant

Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 | PL7 Micro/Junior |
|-----------------------------|-------|------------------|
| Addition | ADDF | + |
| Soustraction | SUBF | - |
| Multiplication | MULF | * |
| Division | DIVF | / |
| Racine carrée | SQRTF | SQRT |
| Valeur absolue | - | ABS |
| Test d'égalité | EQUF | = |
| Test de stricte supériorité | SUPF | > |
| Test de stricte infériorité | INFF | < |
| Autres tests | - | >=, <=, <> |

Instruction sur chaînes d'octets

Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 | PL7 Micro/Junior |
|---------------------|---------|------------------|
| Décalage circulaire | SLCWORD | - |

Instructions de conversion

Tableau des différences entre PL7-2, PL7-3 et PL7-Micro/Junior

| Objets | PL7-2 | PL7-3 | PL7 Micro/Junior |
|----------------------------|-------|-------|------------------------------------|
| Conversion BCD binaire | BCD | DTB | BCD_TO_INT |
| Conversion binaire BCD | BIN | BTD | INT_TO_BCD |
| Conversion ASCII binaire | ATB | ATB | STRING_TO_INT ou STRING_TO_DINT |
| Conversion binaire ASCII | BTA | BTA | INT_TO_STRING ou DINT_TO_STRING |
| Conversion Gray binaire | - | GTB | GRAY_TO_INT |
| Conversion flottant entier | - | FTB | REAL_TO_INT ou REAL_TO_DINT |
| Conversion entier flottant | - | FTF | INT_TO_REAL ou DINT_TO_REAL |
| Conversion BCD flottant | - | DTF | BCD_TO_REAL |

| Objets | PL7-2 | PL7-3 | PL7 Micro/Junior |
|---------------------------|-------|-------|------------------|
| Conversion flottant BCD | - | FTD | REAL_TO_BCD |
| Conversion ASCII flottant | - | ATF | STRING_TO_REAL |
| Conversion flottant ASCII | - | FTA | REAL_TO_STRING |

Instructions sur tableaux

Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 | PL7 Micro/Junior |
|--------------------------------|-----------------|-------------------|
| Opérations arithmétiques | +, -, *, /, REM | +, -, *, /, REM |
| Opérations logiques | AND, OR, XOR | AND, OR, XOR, NOT |
| Addition des mots d'un tableau | + | SUM |
| Recherche du 1er mot différent | EQUAL | EQUAL |
| Recherche du 1er mot égal | SEARCH | FIND_EQU |

Instructions sur programme

Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 | PL7 Micro/Junior |
|--------------------------|--------------|---------------------|
| Saut | JUMP Li | JUMP %Li |
| Appel de sous-programme | - | CALL SRi SRi |
| Retour de sous-programme | RET | RETURN |
| Arrêt de l'application | HALT | HALT |
| Phrase conditionnelle | IF/THEN/ELSE | IF/THEN/ELSE/END_IF |
| Phrase itérative | WHILE/DO | WHILE/DO/END_WHILE |

Instructions sur interruptions

Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 | PL7 Micro/Junior |
|-----------------------------------|----------|------------------|
| Test | READINT | - |
| Masquage | MASKINT | MASKEVT |
| Démasquage | DMASKINT | UNMASKEVT |
| Acquittement | ACKINT | - |
| Génération d'une IT vers coupleur | SETIT | - |

**Instructions d'E/
S explicites**

Tableau des différences entre PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 | PL7 Micro/Junior |
|--------------------------|----------|------------------|
| Lecture des entrées TOR | READBIT | - |
| Ecriture des sorties TOR | WRITEBIT | - |
| Lecture des registres | READREG | - |
| Ecriture des registres | WRITEREG | - |
| Lecture de mots | READEXT | - |
| Ecriture de mots | WRITEEXT | - |

**Instructions sur
blocs fonction**

Tableau des différences entre PL7-2, PL7-3 et PL7-Micro/Junior

| Objets | PL7-3 | PL7 Micro/Junior |
|---------------------------------|----------------------|------------------|
| Présélection | PRESET Ti / Ci | PRESET %Ti / %Ci |
| Lancement | START Ti / Mi | START %Ti / %MNi |
| Activation de la tâche | START CTRLi | - |
| Remise à zéro | RESET Ci / Ri / TXTi | RESET %Ci / %Ri |
| Désactivation de la tâche | RESET CTRLi | - |
| Comptage | UP Ci | UP %Ci |
| Décomptage | DOWN Ci | DOWN %Ci |
| Stockage dans un registre | PUT Ri | PUT %Ri |
| Déstockage d'un registre | GET Ri | GET %Ri |
| Réception d'un message | INPUT TXTi | - |
| Emission d'un message | OUTPUT TXTi | - |
| Emission/Réception d'un message | EXCHG TXTi | - |
| Exécution d'un OFB | EXEC < OFBi > | - |
| Lecture de télégrammes | READTLG | - |

Délimiteurs

Différences

Tableau des différences entre PL7-2/3 et PL7-Micro/Junior

| Objets | PL7-2/3 | PL7 Micro/Junior |
|-------------------------------------|--------------|------------------|
| Affectation | -> | := |
| Parenthèse gauche pour l'indexation | (| [|
| Parenthèse droite pour l'indexation |) |] |
| Longueur de tableau | [longueur] | : longueur |

Aide-mémoire

2

Présentation

Contenu de ce chapitre

Ce chapitre contient un aide-mémoire des instructions du langage PL7

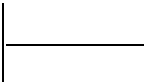
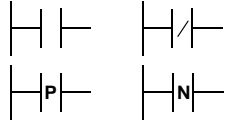
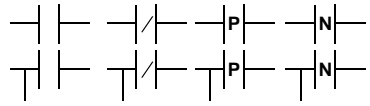


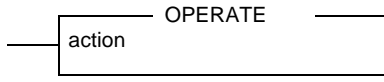
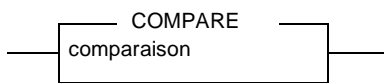
Contenu de ce chapitre

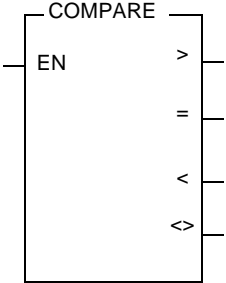
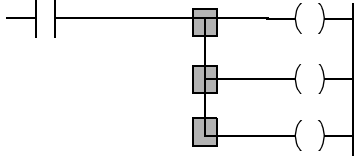
Ce chapitre contient les sujets suivants :

| Sujet | Page |
|--|------|
| Instructions booléennes | 28 |
| Instructions ST | 30 |
| Blocs fonctions LD et IL | 31 |
| Blocs fonctions ST | 32 |
| Structures de contrôle ST | 33 |
| Arithmétique entière (simple et double longueur) | 34 |
| Arithmétique sur flottants | 35 |
| Conversions numériques | 36 |
| Tableaux de bits | 37 |
| Instructions sur tableaux | 38 |
| Instructions sur tableaux de flottants | 39 |
| Instructions "Orphée" | 40 |
| Echanges explicites | 41 |
| Instructions de gestion du temps | 42 |
| Instructions de temporisation | 43 |
| Instructions d'archivage de données | 44 |
| Instructions sur chaînes de caractères | 45 |
| Multi-tâches et événements | 46 |
| Communication | 47 |

Instructions booléennes

Aide-mémoire Aide mémoire rapide des instructions booléennes

| Instructions booléennes | LD | IL |
|--|---|---|
| Initialisation accumulateur ou réseau |  | LD TRUE |
| Test (lecture) direct, inverse, front montant, front descendant |  | LD LDN LDR LDF |
| Et logique |  | AND ANDN ANDR ANDF AND(AND(N AND(R AND(F |
| Inversion | - | N |
| Ou logique exclusif (direct, inverse, front montant, front descendant) | - | XOR XORN XORR XORF |
| Ecriture (directe, inverse) |  | ST STN |
| Mise à 1 Mise à 0 |  | S R |
| Bloc opération (contenu: voir pages suivantes) |  | [action] |
| Bloc comparaison horizontal (contenu: voir pages suivantes) |  | LD [comparaison] AND [comparaison] AND([comparaison] OR [comparaison] OR([comparaison] XOR [comparaison] |

| Instructions booléennes | LD | IL |
|---------------------------------------|---|-------------------|
| Bloc comparaison verticale |  | - |
| MemoryPusH MemoryReaD MemoryPOP |  | MPS MRD MPP |

Instructions ST

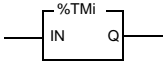
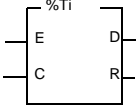
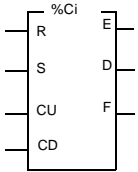
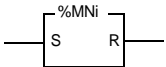
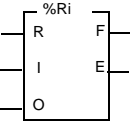
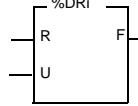
Aide-mémoire

Aide mémoire rapide des instructions ST

| Instructions | ST |
|---------------------------|-------------|
| Affectation | : = |
| OU booléen | OR |
| ET booléen | AND |
| OU exclusif booléen | XOR |
| Inversion | NOT |
| Front montant, descendant | RE , FE |
| Mise à 1, mise à 0 | SET , RESET |

Blocs fonctions LD et IL

Aide-mémoire Aide mémoire rapide des Blocs fonctions LD et IL

| Blocs fonctions | LD | IL |
|---------------------|---|---|
| Temporisateurs IEC |  | IN structure BLK..END_BLK |
| Temporisateur PL-3 |  | - |
| Compteur-décompteur |  | R S CU CD structure BLK..END_BLK |
| Monostable |  | S structure BLK..END_BLK |
| Registre |  | R I O STN structure BLK..END_BLK |
| Drum |  | R U structure BLK..END_BLK |

Blocs fonctions ST

Aide-mémoire

Aide mémoire rapide des iblocs fonctions ST

| Blocs fonctions | ST |
|---------------------|--|
| Temporisateur IEC | START%TMi DOWN%TMi |
| Temporisateur PL-3 | PRESET%Ti START%Ti STOP%Ti |
| Compteur-décompteur | RESET%Ci PRESET%Ci UP%Ci , DOWN%Ci |
| Monostable | START%MNi |
| Registre | RESET%Ri PUT%Ri GET%Ri |
| Drum | RESET%DRi UP%DRi |

Structures de contrôle ST

Aide-mémoire

Aide mémoire rapide des structures de contrôle ST

| Structures de contrôle | ST |
|---------------------------------|---|
| Action conditionnelle | IF... THEN... ELSIF... THEN.. ELSE... END_IF; |
| Action itérative conditionnelle | WHILE... DO... END_WHILE; |
| Action itérative conditionnelle | REPEAT... UNTIL... END_REPEAT; |
| Action répétitive | FOR... DO... END_FOR; |
| Instruction de sortie de boucle | EXIT |

Arithmétique entière (simple et double longueur)

Aide-mémoire

Aide mémoire rapide des instructions arithmétiques entières

| Arithmétique entière (simple et double longueur) | LD/IL/ST |
|--|----------------|
| Transfert ou initialisation | := |
| Comparaisons | = <> <= < > >= |
| Addition, soustraction, multiplication, division, reste de la division | + - * / REM |
| ET, OU, OU exclusif, complément | AND OR XOR NOT |
| Valeur absolue, Racine carrée | ABS, SQRT |
| Incrément | INC |
| Décrément | DEC |
| Décalage à gauche | SHL |
| Décalage à droite | SHR |
| Décalage circulaire à gauche | ROL |
| Décalage circulaire à droite | ROR |

Arithmétique sur flottants

Aide-mémoire Aide mémoire rapide des instructions arithmétiques sur flottants

| Arithmétiques sur flottants | LD/IL/ST |
|--|------------------------|
| Transfert ou initialisation | := |
| Comparaisons | = <> <= < > >= |
| Addition, soustraction, multiplication, division, partie entière | + - * / TRUNC |
| Valeur absolue, racine carrée | ABS, SQRT |
| Logarithme, exponentielles | LOG, LN, EXPT, EXP |
| Sinus, cosinus, tangente | SIN, COS, TAN |
| Arc sinus, arc cosinus, arc tangente | ASIN, ACOS, ATAN |
| Conversion degré <--> radian | DEG_TO_RAD, RAD_TO_DEG |

Conversions numériques

Aide-mémoire Aide mémoire rapide des instructions de conversions numériques

| Conversions numériques | LD/IL/ST |
|---|--------------|
| Conversion BCD en entier simple longueur | BCD_TO_INT |
| Conversion GRAY en entier simple longueur | GRAY_TO_INT |
| Conversion entier simple longueur en BCD | INT_TO_BCD |
| Conversion entier simple longueur en flottant | INT_TO_REAL |
| Conversion entier double longueur en flottant | DINT_TO_REAL |
| Conversion flottant en entier simple longueur | REAL_TO_INT |
| Conversion flottant en entier double longueur | REAL_TO_DINT |
| Conversion BCD 32 bits en entier 32 bits | DBCD_TO_DINT |
| Conversion entier 32 bits en BCD 32 bits | DINT_TO_DBCD |
| Conversion BCD 32 bits en entier 16 bits | DBCD_TO_INT |
| Conversion entier 16 bits en BCD 32 bits | INT_TO_DBCD |
| Extraction d'un mot de poids faible d'un mot double | LW |
| Extraction d'un mot de poids fort d'un mot double | HW |
| Concaténation de 2 mots simples | CONCATW |

Tableaux de bits

Aide-mémoire Aide mémoire rapide des instructions sur tableaux de bits

| Tableaux de bits | LD/IL/ST |
|--|------------|
| Transfert ou intialisation | : = |
| Copie d'un tableau de bits dans un tableau de bits | COPY_BIT |
| ET entre deux tableaux | AND_ARX |
| OU entre deux tableaux | OR_ARX |
| OU exclusif entre deux tableaux | XOR_ARX |
| Négation sur un tableau | NOT_ARX |
| Copie d'un tableau de bits dans un tableau de mots | BIT_W |
| Copie d'un tableau de bits dans un tableau de doubles mots | BIT_D |
| Copie d'un tableau de mots dans un tableau de bits | W_BIT |
| Copie d'un tableau de doubles mots dans un tableau de bits | D_BIT |
| Calcul de la longueur d'un tableau | LENGTH_ARX |

Instructions sur tableaux

Aide-mémoire Aide mémoire rapide des instructions sur tableaux

| Instructions sur tableaux | LD/IL/ST |
|--|------------------------|
| Transfert ou initialisation | : = |
| Opérations arithmétiques entre tableaux | + - * / REM |
| Opérations logiques entre tableaux | AND OR XOR |
| Opérations arithmétiques entre un tableau et un entier | + - * / REM |
| Opérations logiques entre un tableau et un entier | AND OR XOR |
| Complément des éléments d'un tableau | NOT |
| Somme de tous les éléments d'un tableau | SUM |
| Comparaison de deux tableaux | EQUAL |
| Recherche du 1er élément d'un tableau égal à une valeur | FIND_EQW, FIND_EQD |
| Recherche du 1er élément d'un tableau égal à une valeur depuis un rang | FIND_EQWP, FIND_EQDP |
| Recherche du 1er élément d'un tableau sup.à une valeur | FIND_GTW, FIND_GTD |
| Recherche du 1er élément d'un tableau inf.à une valeur | FIND_LTW, FIND_LTD |
| Recherche de la plus grande valeur dans un tableau | MAX_ARW, MAX_ARD |
| Recherche de la plus petite valeur dans un tableau | MIN_ARW, MIN_ARD |
| Nombre d'occurrences d'une valeur dans un tableau | OCCUR_ARW, OCCUR_ARD |
| Décalage circulaire à gauche d'un tableau | ROL_ARW, ROL_ARD |
| Décalage circulaire à droite d'un tableau | ROR_ARW, ROR_ARD |
| Tri d'un tableau (croissant ou décroissant) | SORT_ARW, SORT_ARD |
| Calcul de la longueur d'un tableau | LENGTH_ARW, LENGTH_ARD |

Instructions sur tableaux de flottants

Aide-mémoire Aide mémoire rapide des instructions sur tableaux de flottants

| Instructions sur tableaux de flottants | LD/IL/ST |
|---|------------|
| Transfert et initialisation | : = |
| Somme de tous les éléments d'un tableau | SUM_ARR |
| Comparaison de deux tableaux | EQUAL_ARR |
| Recherche du 1er élément d'un tableau égal à une valeur | FIND_EQR |
| Recherche du 1er élément d'un tableau sup.à une valeur | FIND_GTR |
| Recherche du 1er élément d'un tableau inf.à une valeur | FIND_LTR |
| Recherche de la plus grande valeur dans un tableau | MAX_ARR |
| Recherche de la plus petite valeur dans un tableau | MIN_ARR |
| Nombre d'occurrences d'une valeur dans un tableau | OCCUR_ARR |
| Décalage circulaire à gauche d'un tableau | ROL_ARR |
| Décalage circulaire à droite d'un tableau | ROR_ARR |
| Tri d'un tableau (croissant ou décroissant) | SORT_ARR |
| Calcul de la longueur d'un tableau | LENGHT_ARR |

Instructions "Orphée"

Aide-mémoire Aide mémoire rapide des instructions "Orphée"

| Instructions "Orphée" | LD/IL/ST |
|--|----------------------|
| Décalage à gauche sur mot avec récupération des bits décalés | WSHL_RBIT, DSHL_RBIT |
| Décalage à droite sur mot avec extension de signe et récupération des bits décalés | WSHR_RBIT, DSHR_RBIT |
| Décalage à droite sur mot avec remplissage par 0 et récupération des bits décalés | WSHRZ_C, DSHRZ_C |
| Comptage/décomptage avec signalisation de dépassement | SCOUNT |
| Décalage circulaire gauche | ROLW, ROLD |
| Décalage circulaire droit | RORW, RORD |

Echanges explicites

Aide-mémoire Aide mémoire rapide des Echanges explicites

| Echanges explicites | LD/IL/ST |
|--|-----------------|
| Lecture des paramètres %M d'une voie logique | READ_PARAM |
| Lecture des %M de status d'une voie logique | READ_STS |
| Restitution des paramètres %M d'une voie logique | RESTORE_PARAM |
| Sauvegarde des paramètres %M d'une voie logique | SAVE_PARAM |
| Ecriture des %M de commande d'une voie logique | WRITE_CMD |
| Ecriture des paramètres %M d'une voie logique | WRITE_PARAM |

Instructions de gestion du temps

Aide-mémoire Aide mémoire rapide des instructions de gestion du temps

| Instructions de gestion du temps | LD/IL/ST |
|--|----------------|
| Horodateur | SCHEDULE |
| Comparaisons | = <> <= < > >= |
| Transfert | := |
| Lecture de la date et du code du dernier arrêt de l'automate | PTC |
| Lecture de la date système | RRTC |
| Mise à jour de la date système | WRTC |
| Ajout d'une durée à une date complète | ADD_DT |
| Ajout d'une durée à une heure du jour | ADD_TOD |
| Conversion d'une date en chaîne | DATE_TO_STRING |
| Jour de la semaine | DAY_OF_WEEK |
| Ecart entre deux dates | DELTA_D |
| Ecart entre deux dates complètes | DELTA_DT |
| Ecart entre deux heures du jour | DELTA_TOD |
| Conversion d'une date complète en chaîne | DT_TO_STRING |
| Retrait d'une durée à une date complète | SUB_DT |
| Retrait d'une durée à une heure du jour | SUB_TOD |
| Conversion d'une durée en chaîne | TIME_TO_STRING |
| Conversion d'une heure du jour en chaîne | TOD_TO_STRING |
| Mise sous la forme heures-mn-sec d'une durée | TRANS_TIME |

Instructions de temporisation

Aide-mémoire Aide mémoire rapide des instructions de temporisation

| Instructions de temporisation | LD/IL/ST |
|--------------------------------------|----------|
| Temporisation au déclenchement | FTON |
| Temporisation au déclenchement | FTOF |
| Temporisation d'impulsion | FTP |
| Générateur de signaux rectangulaires | FPULSOR |

Instructions d'archivage de données

Aide-mémoire Aide mémoire rapide des Instructions d'archivage de données

| Instructions d'archivage de données | LD/IL/ST |
|--|---------------|
| Initialisation zone d'archivage sur carte PCMCIA | SET_PCMCIA |
| Ecriture de données sur carte PCMCIA | WRITE_PCMCIA |
| Lecture de données sur carte PCMCIA | READ_PCMCIA |
| Initialisation zone d'archivage sur carte DATA Archiving | SET_PCM_EXT |
| Ecriture de données sur carte DATA Archiving | WRITE_PCM_EXT |
| Lecture de données sur carte DATA Archiving | READ_PCM_EXT |

Instructions sur chaînes de caractères

Aide-mémoire Aide mémoire rapide des Instructions sur chaînes de caractères

| Instructions sur chaînes de caractères | LD/IL/ST |
|---|----------------|
| Comparaisons | = <> <= < > >= |
| Transfert | : = |
| Conversion d'un entier double en string | DINT_TO_STRING |
| Conversion d'un entier simple en string | INT_TO_STRING |
| Conversion d'une chaîne en entier double | STRING_TO_DINT |
| Conversion d'une chaîne en entier simple | STRING_TO_INT |
| Conversion d'une chaîne en flottant | STRING_TO_REAL |
| Conversion d'un flottant en string | REAL_TO_STRING |
| Concaténation de deux chaînes | CONCAT |
| Suppression d'une sous-chaîne | DELETE |
| Recherche du premier caractère différent | EQUAL_STR |
| Recherche d'une sous-chaîne | FIND |
| Insertion d'une sous-chaîne | INSERT |
| Extraction de la partie gauche d'une chaîne | LEFT |
| Longueur d'une chaîne | LEN |
| Extraction d'une sous-chaîne | MID |
| Remplacement d'une sous-chaîne | REPLACE |
| Extraction de la partie droite d'une chaîne | RIGHT |

Multi-tâches et événements

Aide-mémoire Aide mémoire rapide des Instructions de Multi-tâches et événements

| Multi-tâches et événements | LD/IL/ST |
|--|---------------------|
| Activation / désactivation d'une tâche | positionnement %Si |
| Régler le temps de cycle d'une tâche | positionnement %SWi |
| Masquage global des événements | MASKEVT |
| Démasquage global des événements | UNMASKEVT |

Communication

Aide-mémoire Aide mémoire rapide des Instructions de communication

| Communication | LD/IL/ST |
|--|-------------|
| Demande d'arrêt d'une fonction en cours | CANCEL |
| Emission de données et/ou réception de données | DATA_EXCH |
| Demande de lecture d'une chaîne de caractères | INPUT_CHAR |
| Emission et/ou demande de réception d'une chaîne de caractères | OUT_IN_CHAR |
| Emission d'une chaîne de caractères | PRINT_CHAR |
| Réception d'un télégramme | RCV_TLG |
| Lecture d'objets langage de base | READ_VAR |
| Emission/réception de requêtes UNI-TE | SEND_REQ |
| Emission d'un télégramme | SEND_TLG |
| Ecriture d'objets langage de base | WRITE_VAR |
| Décalage d'un octet à droite dans un tableau | ROR1_ARB |
| Permutation des octets dans un tableau de mots | SWAP |
| Lecture de données communes Modbus+ | READ_GDATA |
| Ecriture de données communes Modbus+ | WRITE_GDATA |
| Gestion de la connexion carte modem | CALL_MODEM |
| Serveur immédiat | SERVER |
| Ecriture 1 K messagerie | WRITE_ASYN |
| Lecture 1 K messagerie | READ_ASYN |

Mots réservés

3

Mots réservés

Liste de mots réservés

Les mots réservés ci-après ne doivent pas être utilisés en tant que symboles.

Mots de A à C

Liste de mots réservés

| | | | |
|----------------|-------------|-----------|----------------|
| ABS | ANY_REAL | BLOCK | CLOSED_CONTACT |
| ACCEPT | ARRAY | BODY | COIL |
| ACOS | AR_D | BOOL | COMMAND |
| ACTION | AR_DINT | BOTTOM | COMMENTS |
| ACTIVATE_PULSE | AR_F | BTI | COMP4 |
| ACTIVE_TIME | AR_INT | BTR | COMPCH |
| ADD | AR_R | BY | CONCAT |
| ADDRESS | AR_W | BYTE | CONCATW |
| ADD_DT | AR_X | C | CONF |
| ADD_TOD | ASIN | CAL | CONFIGURATION |
| ADR | ASK | CALC | CONSTANT |
| AND | ASK_MSG | CALCN | CONTROL_LEDS |
| ANDF | ASK_VALUE | CALL | COPY_BIT |
| ANDN | ASSIGN_KEYS | CALL_COIL | COS |
| ANDR | AT | CANCEL | CTD |
| AND_ARX | ATAN | CASE | CTU |
| ANY | AUX | CD | CTUD |
| ANY_BIT | BCD_TO_INT | CHART | CU |
| ANY_DATE | BIT_D | CH_M | |
| ANY_INT | BIT_W | CLK | |
| ANY_NUM | BLK | CLOSE | |

Mots de D à E Liste de mots réservés

| | | | |
|------------------|------------|--------------------|----------------|
| D | DO | END | END_REPEAT |
| DATE | DOWN | ENDC | END_RESOURCE |
| DATE_AND_TIME | DR | ENDCN | END_RUNG |
| DAT_FMT | DRUM | END_ACTION | END_STEP |
| DAY_OF_WEEK | DS | END_BLK | END_STRUCT |
| DA_TYPE | DSHL_RBIT | END_BLOCK | END_TRANSITION |
| DEACTIVATE_PULSE | DSHRZ_C | END_CASE | END_TYPE |
| DEC | DSHR_RBIT | END_COMMENTS | END_VAR |
| DELETE | DSORT_ARD | END_CONFIGURATION | END_WHILE |
| DELTA_D | DSORT_ARW | END_FOR | EQ |
| DELTA_DT | DT | END_FUNCTION | EQUAL |
| DELTA_TOD | DTS | END_FUNCTION_BLOCK | EQUAL_ARR |
| DINT | DWORD | END_IF | ERR |
| DINT_TO_REAL | D_BIT | END_MACRO_STEP | EVT |
| DINT_TO_STRING | E | END_PAGE | EXCHG |
| DISPLAY_ALRM | EBOOL | END_PHRASE | EXCH_DATA |
| DISPLAY_GRP | ELSE | END_PROG | EXIT |
| DISPLAY_MSG | ELSIF | END_PROGRAM | EXP |
| DIV | EMPTY | | EXPT |
| DMOVE | EMPTY_LINE | | |

Mots de F à J Liste de mots réservés

| | | | |
|-----------|----------------|-------------|---------------|
| F | FIND_LTW | GR7 | INFO |
| FALSE | FOR | GRAY_TO_INT | INITIAL_STEP |
| FAST | FPULSOR | GT | INIT_BUTTONS |
| FBD | FROM | GTI | INPUT |
| FE | FTOF | H | INPUT_CHAR |
| FIFO | FTON | HALT | INSERT |
| FIND | FTP | HALT_COIL | INT |
| FIND_EQ | FUNC | HASH_COIL | INTERVAL |
| FIND_EQD | FUNCTION | HW | INT_TO_BCD |
| FIND_EQDP | FUNCTION_BLOCK | H_COMPARE | INT_TO_REAL |
| FIND_EQR | F_B | H_LINK | INT_TO_STRING |
| FIND_EQW | F_EDGE | I | ITB |
| FIND_EQWP | F_TRIG | IF | ITS |
| FIND_GTD | GE | IL | JMP |
| FIND_GTR | GET | IN | JMPC |
| FIND_GTW | GET_MSG | INC | JMPCN |
| FIND_LTD | GET_VALUE | INCJUMP | JUMP |
| FIND_LTR | GLOBAL_COMMENT | INDEX_CH | JUMP_COIL |

Mots de L à M Liste de mots réservés

| | | | |
|------------|------------|-----------|--------------|
| L | LIFO | MASKEVT | MIN_ARR |
| LAD | LIMIT | MAST | MIN_ARW |
| LANGAGE | LINT | MAX | MN |
| LANGUAGE | LIST | MAX_ARD | MOD |
| LD | LIT | MAX_ARR | MONO |
| LDF | LN | MAX_ARW | MOVE |
| LDN | LOCATION | MAX_PAGES | MPP |
| LDR | LOG | MAX_STEP | MPS |
| LE | LREAL | MCR | MRD |
| LEFT | LT | MCR_COIL | MS |
| LEN | LW | MCS | MUL |
| LENGTH_ARD | LWORD | MCS_COIL | MUX |
| LENGTH_ARR | M | MID | M_CH |
| LENGTH_ARW | MACRO_STEP | MIN | M_MACRO_STEP |
| LENGTH_ARX | MAIN | MIN_ARD | |

Mots de N à P Liste de mots réservés

| | | | |
|-------------------|----------------|----------------|---------------|
| N | NB_TRANSITIONS | OPEN_CONTACT | PID |
| N1 | NE | OPERATE | PID_MMI |
| NAME | NIL | OR | PLC |
| NB_ACTIVE_STEPS | NO | ORF | POST |
| NB_ACTIVE_TIME | NON_STORED | ORN | PRESET |
| NB_BLOCKS | NOP | ORR | PRINT |
| NB_COMMON_WORDS | NOT | OR_ARX | PRINT_CHAR |
| NB_CONSTANT_WORDS | NOT_ARX | OTHERS | PRI00 |
| NB_CPT | NOT_COIL | OUT | PRI01 |
| NB_DRUM | NOT_READABLE | OUTIN_CHAR | PRIORITY |
| NB_INTERNAL_BITS | NO_GR7 | OUTPUT | PRL |
| NB_INTERNAL_WORDS | NO_PERIOD | OUT_BLK | PROG |
| NB_MACRO_STEPS | N_CONTACT | P | PROGRAM |
| NB_MONO | O | P0 | PROG_LANGAGE |
| NB_PAGES | OCCUR | P1 | PROG_LANGUAGE |
| NB_REG | OCCUR_ARD | PAGE | PT |
| NB_TIMER | OCCUR_ARR | PAGE_COMMENT | PTC |
| NB_TM | OCCUR_ARW | PANEL_CMD | PUT |
| | OF | PERIOD | PV |
| | ON | PHRASE | PWM |
| | OPEN | PHRASE_COMMENT | P_CONTACT |

Mots de Q à R

Liste de mots réservés

| | | | |
|--------------|----------------|-----------|-----------|
| Q | REAL_TO_INT | RETURN | ROR_ARR |
| QUERY | REAL_TO_STRING | RET_COIL | ROR_ARW |
| R | REG | RIGHT | ROR_DWORD |
| R1 | REM | ROL | ROR_WORD |
| RCV_TLG | REPEAT | ROLD | RRTC |
| RE | REPLACE | ROLW | RS |
| READ | REQ | ROL_ARR | RTB |
| READ_EVT_UTW | RESET | ROL_ARR | RTC |
| READ_ONLY | RESET_COIL | ROL_ARW | RTS |
| READ_PARAM | RESOURCE | ROL_DWORD | RUNG |
| READ_STS | RESTORE_PARAM | ROL_WORD | R_EDGE |
| READ_VAR | RET | ROR | R_TRIG |
| READ_WRITE | RETAIN | RORD | |
| REAL | RETC | RORW | |
| REAL_TO_DINT | RETCN | ROR_ARR | |

Mots de S à S

Liste de mots réservés

| | | | |
|----------------|------------|--------------------|----------------|
| S | SEND_REQ | SL | STN |
| S1 | SEND_TLG | SLCWORD | STOP |
| SAVE | SERVO | SMOVE | STR |
| SAVE_PARAM | SET | SOFT_CONFIGURATION | STRING |
| SCHEDULE | SET_COIL | SORT | STRING_TO_DINT |
| SD | SFC | SORT_ARR | STRING_TO_INT |
| SEARCH | SHIFT | SORT_ARR | STRING_TO_REAL |
| SECTION | SHL | SORT_ARW | STRUCT |
| SEL | SHOW_ALARM | SQRT | SUB |
| SEMA | SHOW_MSG | SR | SUB_DT |
| SEND | SHOW_PAGE | ST | SUB_TOD |
| SENDER | SHR | STANDARD | SUM |
| SEND_ALARM | SHRZ | START | SU_TYPE |
| SEND_MBX_ALARM | SIN | STD | SWAP |
| SEND_MBX_MSG | SINGLE | STEP | S_T_AND_LINK |
| SEND_MSG | SINT | STI | S_T_OR_LINK |

Mots T à W

Liste de mots réservés

| | | | |
|-------------|--------------|--------------|-------------|
| T | TOP | UP | W |
| TAN | TP | USINT | WHILE |
| TASK | TRANSITION | USORT_ARD | WITH |
| TASKS | TRANS_TIME | USORT_ARW | WORD |
| THEN | TRUE | UTIN_CHAR | WRITE |
| TIME | TRUNC | VAR | WRITE_CMD |
| TIMER | TYPE | VAR_ACCESS | WRITE_PARAM |
| TIME_OF_DAY | TYPES | VAR_EXTERNAL | WRITE_VAR |
| TM | T_S_AND_LINK | VAR_GLOBAL | WRTC |
| TMAX | T_S_OR_LINK | VAR_INPUT | WSHL_RBIT |
| TMOVE | U | VAR_IN_OUT | WSHRZ_C |
| TO | UDINT | VAR_OUTPUT | WSHR_RBIT |
| TOD | UINT | VAR_PUBLIC | W_BIT |
| TOF | ULINT | VERSION | |
| TOFF | UNMASKEVT | V_COMPARE | |
| TON | UNTIL | V_LINK | |

Mots X, Y et Divers

Liste de mots réservés

| | |
|----------|-------------------|
| XM | *_TO_* * = Letter |
| XM_MONO | SRi |
| XM_MULTI | AUXi |
| XOR | EVTi |
| XORF | XMi |
| XORN | i = entier |
| XORR | |
| XOR_ARX | |
| YES | |

Conformité au standard CEI1131-3

4

Présentation

Contenu de ce chapitre

Ce chapitre décrit la conformité au standard CEI1131-3 : "Automates Programmables"

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

| Sujet | Page |
|--|------|
| Conformité au standard CEI 1131-3 | 56 |
| Eléments commun | 57 |
| Eléments de langage IL | 63 |
| Eléments de langage ST | 64 |
| Eléments graphiques communs | 65 |
| Eléments de langage LD | 66 |
| Paramètres dépendant de l'implémentation | 67 |
| Situations d'erreurs | 70 |

Conformité au standard CEI 1131-3

Présentation du standard CEI 1131-3

Le standard CEI 1131-3 «Automates programmables - Partie 3: Langages de programmation» spécifie la syntaxe et la sémantique des éléments logiciels mis en oeuvre pour programmer les automates programmables.

Ce standard comporte la description de 2 langages textuels, IL (Instruction List) et ST (Structured Text), de 2 langages graphiques, LD (Ladder Diagram) et FBD (Function Block Diagram) et d'un formalisme graphique, SFC (Sequential Function Chart), qui permet de structurer l'organisation interne d'une séquence programmée.

Le logiciel de programmation PL7 permet de programmer un automate programmable conformément au standard CEI: PL7 implémente un sous-ensemble des éléments de langages définis dans le standard et définit des extensions, extensions qui sont autorisées dans le cadre de ce standard.

Le standard CEI 1131-3 ne définit pas les règles d'interactivité du logiciel fourni par un constructeur se réclamant conforme au standard, ce qui laisse une grande souplesse de présentation et de saisie des éléments de programmation pour le confort de l'utilisateur.

Les éléments du standard implémentés dans PL7, les informations spécifiques d'implémentation et les cas de détection d'erreurs sont synthétisés dans des tables de conformité.

Eléments commun

Généralités

Description des caractéristiques communes aux langages qui sont conformes à la CEI 1131-3

Le tableau suivant liste en regard des tableaux de caractéristiques décrites dans le standard IEC 1131-3, l'implémentation faite dans les langages PL7. Ceci pour les caractéristiques communes à tous les langages PL7.

Tableau de caractéristiques

Caractéristiques communes à tous les langages:

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|--------------------|---|
| 1 | 1 | Jeu de caractères requis (voir paragraphe 2.1.1 de 1131-3) |
| 1 | 2 | Caractères minuscules |
| 1 | 3a | Signe numéro (#) |
| 1 | 4a | Signe dollar (\$) |
| 1 | 5a | Barre verticale () |
| 1 | 6a | Délimiteurs d'indices : Crochets gauche et droit "[]" |
| 2 | 1 | Majuscules et nombres |
| 2 | 2 | Majuscule et minuscule, nombres, caractères de soulignements intégrés |
| 3 | 1 | Commentaires |
| 4 | 1 | Littéraux entiers (Note 1) |
| 4 | 2 | Littéraux réels (Note 1) |
| 4 | 3 | Littéraux réels avec des exposants |
| 4 | 4 | Littéraux en base 2 (Note 1) |
| 4 | 6 | Littéraux en base 16 (Note 1) |
| 4 | 7 | Booléens Zéro et Un |
| 4 | 8 | Booléens TRUE et FALSE |
| 5 | 1 | Caractéristiques des littéraux chaînes de caractères |
| 6 | 2 | \$\$ Signe dollar |
| 6 | 3 | \$' Apostrophe |
| 6 | 4 | \$L ou \$I Changement de ligne |
| 6 | 5 | \$N ou \$n Nouvelle ligne |
| 6 | 6 | \$P ou \$p Changement de page |
| 6 | 7 | \$R ou \$r Retour du curseur |
| 6 | 8 | \$T ou \$t Tabulation |
| 7 | 1a | Littéraux de durée avec préfixe court t# (Note 2) |

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|--|---|
| 10 | 1 | BOOL -1 bit- |
| 10 | 10 | REAL -32 bits- |
| 10 | 12 | TIME -32 bits- (Note 3) |
| 10 | 13 | DATE -32 bits- (Note 3) |
| 10 | 14 | TIME_OF_DAY -32 bits- (Note 3) |
| 10 | 15 | DATE_AND_TIME -64 bits- (Note 3) |
| 10 | 16 | STRING |
| 10 | 17 | BYTE -8 bits- |
| 10 | 18 | WORD -16 bits- |
| 10 | 19 | DWORD -32 bits- |
| 15 | 1 | Préfixe I pour Input |
| 15 | 2 | Préfixe Q pour Output |
| 15 | 3 | Préfixe M pour Memory |
| 15 | 4 | Préfixe X, taille d'un bit |
| 15 | 5 | Aucun préfixe, taille d'un bit |
| 15 | 6 | Préfixe B, taille d'un octet (8 bits) |
| 15 | 7 | Préfixe W, taille d'un mot (16 bits) |
| 15 | 8 | Préfixe D, taille d'un double mot (32 bits) |
| 16 | VAR VAR_INPUT VAR_OUTPUT VAR_IN_OUT VAR_EXTERNAL VAR_GLOBAL CONSTANT AT | Mots clés (Note 4) |
| 17 | 2 | Déclaration de variables non volatiles directement représentées (repères) (Note 4) |
| 17 | 3 | Déclaration d'emplacements des variables symboliques (symboles ou repères) (Note 4) |
| 17 | 5 | Affectation automatique en mémoire de variables symboliques (variables de blocs fonction) (Note 4) |
| 18 | 2 | Initialisation de variables non volatiles directement représentées (repères) (Note 4) |
| 18 | 3 | Affectation d'emplacements et de valeurs initiales de variables symboliques (symboles sur repères) (Note 4) |
| 18 | 5 | Initialisation de variables symboliques (variables de blocs fonction) (Note 4) |

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|--|---|
| 10 | 17 | BYTE -8 bits- |
| 10 | 18 | WORD -16 bits- |
| 10 | 19 | DWORD -32 bits- |
| 15 | 1 | Préfixe I pour Input |
| 15 | 2 | Préfixe Q pour Output |
| 15 | 3 | Préfixe M pour Memory |
| 15 | 4 | Préfixe X, taille d'un bit |
| 15 | 5 | Aucun préfixe, taille d'un bit |
| 15 | 6 | Préfixe B, taille d'un octet (8 bits) |
| 15 | 7 | Préfixe W, taille d'un mot (16 bits) |
| 15 | 8 | Préfixe D, taille d'un double mot (32 bits) |
| 16 | VAR VAR_INPUT VAR_OUTPUT VAR_IN_OUT VAR_EXTERNAL VAR_GLOBAL CONSTANT AT | Mots clés (Note 4) |
| 17 | 2 | Déclaration de variables non volatiles directement représentées (repères) (Note 4) |
| 17 | 3 | Déclaration d'emplacements des variables symboliques (symboles ou repères) (Note 4) |
| 17 | 5 | Affectation automatique en mémoire de variables symboliques (variables de blocs fonction) (Note 4) |
| 18 | 2 | Initialisation de variables non volatiles directement représentées (repères) (Note 4) |
| 18 | 3 | Affectation d'emplacements et de valeurs initiales de variables symboliques (symboles sur repères) (Note 4) |
| 18 | 5 | Initialisation de variables symboliques (variables de blocs fonction) (Note 4) |

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|--|--|
| 23 | 3 | Fonction LN: logarithme népérien |
| 23 | 4 | Fonction LOG: logarithme en base 10 |
| 23 | 5 | Fonction EXP: exponentielle naturelle |
| 23 | 6 | Fonction SIN: sinus en radian |
| 23 | 7 | Fonction COS: cosinus en radian |
| 23 | 8 | Fonction TAN: tangente en radian |
| 23 | 9 | Fonction ASIN: arc sinus |
| 23 | 10 | Fonction ACOS: arc cosinus |
| 23 | 11 | Fonction ATAN: arc tangente |
| 25 | 1 | Fonction SHL: décalage à gauche |
| 25 | 2 | Fonction SHR: décalage à droite |
| 25 | 3 | Fonction ROR: rotation à droite |
| 25 | 4 | Fonction ROL: rotation à gauche |
| 29 | 1 | Fonction LEN: longueur de chaîne |
| 29 | 2 | Fonction LEFT: n caractères les plus à gauche |
| 29 | 3 | Fonction RIGHT: n caractères les plus à droite |
| 29 | 4 | Fonction MID: n caractères à partir d'une position donnée |
| 29 | 5 | Fonction CONCAT: concaténation de chaînes (Note 7) |
| 29 | 6 | Fonction INSERT: insérer une chaîne dans une autre |
| 29 | 7 | Fonction DELETE: supprimer des caractères |
| 29 | 8 | Fonction REPLACE: remplacer des caractères autre |
| 29 | 9 | Fonction FIND: rechercher une chaîne dans une autre |
| 32 | Inputread Inputwrite Outputread Outputwrite | (Note 8) |
| 33 | 1 | Qualificatif RETAIN pour les variables internes des blocs fonction.) (Note 9) (Note 4) |
| 33 | 2 | Qualificatif RETAIN pour les sorties des blocs fonction (Note 9) (Note 4) |
| 33 | 4a | Déclaration d'entrées/sorties de blocs fonction (littérale). (Note 4) |
| 37 | 1 | Timer à impulsion: TP (Note 10) |
| 37 | 2a | Timer à enclenchement: TON (Note 10) |
| 37 | 3a | Timer à déclenchement: TOF (Note 10) |
| 38 | schémas temporels | TP, TON, TOF |
| 39 | 19 | Utilisation de variables directement représentées (repères) |

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|--|---|
| 40 | 1 | Etape, forme graphique Note: Un numéro d'étape remplace un identificateur d'étape |
| 40 | 2 | Etape, forme textuelle utilisée dans la forme source du Grafcet uniquement |
| 41 | 1 | Condition de transition en langage ST |
| 41 | 2 | Condition de transition en langage LD |
| 42 | 2l | Déclarations des actions en langage LD |
| 43 | 1 | Bloc d'action |
| | 2 | Blocs d'action concaténés |
| 45 | 2 | Qualificatif d'action N (non mémorisé) |
| 45 | 11 | Qualificatif d'action P1 (Pulse rising edge) |
| 45 | 12 | Qualificatif d'action P0 (Pulse falling edge) |
| 46 | 1 | Simple séquence, alternance d'étape/transition |
| 46 | 2c | Divergence en «ou»: l'utilisateur s'assure que les conditions des transitions sont mutuellement exclusives |
| 46 | 3 | Convergence en «ou» |
| 46 | 4 | Divergence en «et», Convergence en «et» |
| 46 | 5c | Saut de séquence dans une divergence en «ou» |
| 46 | 6c | Boucle de séquence: retour à une précédente étape |
| 46 | 7 | Flèches directionnelles Note: Les flèches directionnelles sont montantes et descendantes |
| 48 | 40 41 42 43 44 45 46 | Le langage Grafcet remplit les conditions pour avoir le niveau minimum de conformité SFC 1131-3 Représentation graphique |
| 49 | 3 | Construction RESOURCE...ON...END_RESOURCE |
| 49 | 5a | Construction TASK périodique dans RESOURCE |
| 49 | 6a | Déclaration PROGRAM avec association PROGRAM-to-TASK |
| 49 | 7 | Déclaration de variables directement représentées dans VAR_GLOBAL |
| 50 | 5b | Ordonnancement préemptif dans le modèle multi-tâches |

Note :

- **Note 1** : Les caractères de soulignement (_) insérés entre les chiffres d'un littéral numérique ne sont pas acceptés.
- **Note 2** : Ces littéraux ne sont visibles que dans le source application, pour exprimer les temps des tâches configurées.
- **Note 3** : Ces types de données ne sont pas encore implémentés de façon visible pour l'utilisateur. Ce tableau précise néanmoins l'occupation mémoire de leur représentation interne.
- **Note 4** : Ces mots clés ne sont utilisés que dans les sources générés par PL7 et par l'outil de conversion d'application PL7-2 et PL7-3.
- **Note 5** : Effets de conversions aux limites:
 - DINT_TO_STRING: Si la chaîne accueillant le résultat est inférieure à 13 caractères, il y a troncature et positionnement de %S15.
 - INT_TO_STRING: Si la chaîne accueillant le résultat est inférieure à 7 caractères, il y a troncature et positionnement de %S15.
 - STRING_TO_DINT et STRING_TO_INT: Si la chaîne n'est pas convertible en entier, le résultat est indéterminé et il y a positionnement de %S18.
 - DATE_TO_STRING: Si la chaîne accueillant le résultat est inférieure à 11 caractères, il y a troncature et positionnement de %S15.
 - DT_TO_STRING: Si la chaîne accueillant le résultat est inférieure à 20 caractères, il y a troncature et positionnement de %S15.
 - TIME_TO_STRING: Si la chaîne accueillant le résultat est inférieure à 15 caractères, il y a troncature et positionnement de %S15.
 - TOD_TO_STRING: Si la chaîne accueillant le résultat est inférieure à 9 caractères, il y a troncature et positionnement de %S15.
 - REAL_TO_STRING: Si la chaîne accueillant le résultat est inférieure à 15 caractères, il y a troncature et positionnement de %S15.
 - STRING_TO_REAL: Si la chaîne n'est pas convertible en réel, le résultat vaut «1.#NAN» (16#FFC0_0000) et il y a positionnement de %S18.
 - REAL_TO_INT: Si le réel n'est pas convertible dans les limites [-32768, +32767], le résultat vaut -32768 et il y a positionnement de %S18 et %SW17:X0.
 - REAL_TO_DINT: Si le réel n'est pas convertible dans les limites [-2147483648,+2147483647], le résultat vaut -2147483648 et il y a positionnement de %S18 et %SW17:X0.
 - INT_TO_REAL: La conversion est toujours possible.
 - DINT_TO_REAL: La conversion est toujours possible.
- **Note 6** : Le type INT n'étant pas formellement implémenté - mais tout de même utilisé -, ces fonctions permettent de changer le format de codage d'un WORD.
- **Note 7** : Limitation de la fonction CONCAT à la concaténation de 2 chaînes.
- **Note 8** : Ce paragraphe s'applique aux blocs fonctions prédéfinis PL7.
- **Note 9** : Le qualificatif RETAIN est implicite.
- **Note 10** : Les timers TP, TON, TOF respectent les diagrammes temporels de la table 38, mais présentent une interface d'E/S différente de la 1131-3.

Eléments de langage IL

Généralités

Description des caractéristiques des éléments de langage IL qui sont conformes à la CEI 1131-3

Le tableau suivant liste en regard des tableaux de caractéristiques décrites dans le standard IEC 1131-3, l'implémentation faite dans les langages PL7.

Tableau de caractéristiques

Caractéristiques des éléments de langage IL:

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|----------------------|---|
| 51 | Champs d'instruction | Etiquette, opérateur, opérande, commentaire |
| 52 | 1 | LD |
| 52 | 2 | ST |
| 52 | 3 | S et R |
| 52 | 4 | AND |
| | 5 | OR |
| | 6 | XOR |
| 52 | 18 | JMP |
| 52 | 20 | RET |
| 52 | 21 |) |
| 53 | 3 | Utilisation d'opérateurs d'entrée pour le lancement de blocs fonction en langage IL |
| 54 | 11 | IN (voir Note) |
| 54 | 12 | IN (voir Note) |
| 54 | 13 | IN (voir Note) |

Note : L'opérateur PT n'est pas implémenté.

Eléments de langage ST

Généralités

Description des caractéristiques des éléments de langage ST qui sont conformes à la CEI 1131-3

Le tableau suivant liste en regard des tableaux de caractéristiques décrites dans le standard IEC 1131-3, l'implémentation faite dans les langages PL7. Ce langage est utilisé à part entière dans les modules ST. Un sous-ensemble ST est aussi utilisé dans les blocs OPERATION et COMPARAISON des langages IL et LD.

Tableau de caractéristiques

Caractéristiques des éléments de langage ST :

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|--------------------|--|
| 55 | 1 | Mise entre parenthèses |
| 55 | 2 | Evaluation de fonction |
| 55 | 3 | - Négation |
| 55 | 4 | NOT Complément |
| 55 | 5 | JMP |
| 55 | 6 7 | * Multiplication / Division |
| 55 | 9 10 | + Addition - Soustraction |
| 55 | 11 | <, >, <=, >= Comparaison |
| 55 | 12 | = Egalité |
| 55 | 13 | <> Inégalité |
| 55 | 15 | AND pour le «et» booléen |
| 55 | 16 | XOR pour le «ou exclusif» booléen |
| 55 | 17 | OR pour le «ou» booléen |
| 56 | 1 | := Affectation |
| 56 | 3 | Structure RETURN |
| 56 | 4 | Structure IF «if... then... elsif... then... else... end_if» |
| 56 | 6 | Structure FOR «for... to... do... end_for» (voir Note) |
| 56 | 7 | Structure WHILE «while... do... end_while» |
| 56 | 8 | Structure REPEAT «repeat ... until... end_repeat» |
| 56 | 9 | Structure EXIT |

Note : Implémentation de la boucle FOR avec un pas implicite de 1 (by 1).

Eléments graphiques communs

Généralités

Description des caractéristiques des éléments graphiques communs qui sont conformes à la CEI 1131-3

Le tableau suivant liste en regard des tableaux de caractéristiques décrites dans le standard IEC 1131-3, l'implémentation faite dans les langages PL7.

Tableau de caractéristiques

Caractéristiques des éléments graphiques communs :

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|--------------------|--|
| 57 | 2 | Lignes horizontales graphiques |
| 57 | 4 | Lignes verticales graphiques |
| 57 | 6 | Jonction ligne horizontale / ligne verticale graphique |
| 57 | 8 | Croisement graphique de lignes sans connexion |
| 57 | 10 | Coins connectés et non connectés graphiques |
| 57 | 12 | Blocs avec lignes connectées graphiques |
| 58 | 2 | Saut inconditionnel en langage LD |
| 58 | 4 | Saut conditionnel en langage LD inconditionnel |
| 58 | 5 | Retour conditionnel en langage LD |
| 58 | 8 | Retour inconditionnel en langage LD |

Éléments de langage LD

Généralités

Description des caractéristiques des éléments de langage LD qui sont conformes à la CEI 1131-3

Le tableau suivant liste en regard des tableaux de caractéristiques décrites dans le standard IEC 1131-3, l'implémentation faite dans les langages PL7.

Tableau de caractéristiques

Caractéristiques des éléments de langage LD :

| Tableau No | Caractéristique No | Description des caractéristiques |
|------------|--------------------|--|
| 59 | 1 | Barre d'alimentation gauche |
| 59 | 2 | Barre d'alimentation droite |
| 60 | 1 | Liaison horizontale |
| 60 | 2 | Liaison verticale |
| 61 | 1 | Contact ouverte |
| 61 | 3 | Contact fermée |
| 61 | 5 | Contact détecteur de transition positive |
| 61 | 7 | Contact détecteur de transition négative |
| 62 | 1 | Bobine |
| 62 | 2 | Bobine négative |
| 62 | 3 | Bobine SET |
| 62 | 4 | Bobine RESET |

Paramètres dépendant de l'implémentation

Généralités

Description des paramètres de PL7 qui dépendent de l'implémentation qui en est faite

Le tableau suivant liste en regard des tableaux de caractéristiques décrites dans le standard IEC 1131-3, l'implémentation faite dans les langages PL7.

Tableau de caractéristiques

Caractéristiques des éléments de langage IL:

| Paramètres | Limitation et comportement de PL7 |
|---|--|
| Procédure de traitement d'erreurs | De nombreuses erreurs sont signalées à l'exécution par positionnement de bits et mots système |
| Caractères nationaux utilisés | ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÑÓÔ ÕÖØÙÚÛÜàáâãääæçèéêëíîïñóôõöøùúûÿ #, \$, |
| Longueur maximale des identificateurs | 32 |
| Longueur maximale du commentaires | 222 |
| Plage des valeurs de durée | Note 1 |
| Plage des valeurs de type TIME | Note 1 |
| Précision de la représentation des secondes dans les types TIME_OF_DAY et DATE_AND_TIME | Note 2 |
| Nombre maximal d'indices de tableau | 1 (Note 3) |
| Taille maximal des tableaux | dépendant de la zone indexée (Note 3) |
| Longueur maximale par défaut des variables STRING | ne s'applique pas |
| Longueur maximale autorisée pour les variables STRING | 255 |
| Nombre maximal de niveaux hiérarchiques Configuration logique ou physique | 3 Configuration logique |
| Intervalle maximum des valeurs d'indices | dépendant de la zone indexée (Note 3) |
| Initialisation des entrées du système | Les variables sont initialisées par le système: <ul style="list-style-type: none"> à la valeur initiale spécifiée par l'utilisateur, le cas échéant sinon à zéro |
| Effets des conversions de types sur la précision | cf table 22, caractéristique 1 |
| Nombre maximum de types et d'instances de blocs fonction | Pas de nombre maximum (les limites sont liées au volume de l'application) |

| Paramètres | Limitation et comportement de PL7 |
|---|--|
| Limite de la taille des programmes | Volume maximum de code d'un programme = 64 KOctets |
| Précision sur le temps écoulé associé à une étape | 100ms |
| Nombre maximum d'étapes par graphe | 96 sur automate 3710 128 sur automate 3720 1024 sur automates 57xx V3.0 |
| Nombre maximum de transitions par graphe et par étape) | 1024 transitions par graphe 11 transitions par étape1 |
| Mécanisme de commande des actions | Qualificatifs P0, P1 et N1 |
| Nombre maximum de blocs action par étape | 3 actions sont possibles: à l'activation (P1), continue (N1) et à la désactivation (P0) |
| Indication graphique de l'état d'une étape | Étape active en inversion vidéo |
| Temps de franchissement d'une transition (désactivation des étapes amont et activation des étapes aval) | Le temps de franchissement est variable et toujours non nul |
| Profondeur des constructions divergentes et convergentes | Limite donnée par la grille de saisie |
| Liste des automates programmables par PL7 | TSX MICRO, TSX PREMIUM |
| Nombre maximal de tâches | 1 tâche périodique ou cyclique 1 tâche périodique 8 tâches événementielles pour automates 37 10 16 tâches événementielles pour automates 37 20 32 tâches événementielles pour automates 57 10 64 tâches événementielles pour automates 57 20/30 de 1 ms à 255 ms |
| Plages des intervalles des tâches Ordonnancement préemptif ou non-préemptif | Ordonnancement préemptifs |
| Longueur maximale d'une expression | variable |
| Evaluation partielle des expressions booléennes | non |
| Longueur maximum des structures de commandes en ST | variable |
| Valeur de la variable de commande après exécution complète d'une boucle FOR | La valeur de la variable de commande vaut la valeur limite + 1 (car le pas est de 1) |
| Représentation graphique/semi-graphique Restriction sur la topologie des réseaux | Représentation graphiques Un réseau LD peut s'étaler sur 16 colonnes et 7 lignes au maximum |

Note :

- **Note 1:** Ces types de données ne sont pas encore implémentés de façon visible pour l'utilisateur. Ce tableau précise néanmoins leurs plages de va-leurs dans le format IEC 1131-3.
TIME: de T#0 à T#429496729.5s
TIME_OF_DAY: de TOD#0:0:0 à TOD#23:59:59
DATE_AND_TIME: de DT#1990-01-01:0:0:0 à DT#2099-12-31:23:59:59
DATE: de D#1990-01-01 à D#2099-12-31DT#2099-12-3
- **Note 2:** Les arrondis sont faits de la façon suivante: de x.0 s à x.4 s, on arrondit à x s et de x.5 s à x.9 s on arrondit à x+1 s.
- **Note 3:** Il est possible d'indexer positivement et négativement tous les types de variables directement représentées dans la limite de leur nombre maximal respectif défini en configuration.

Situations d'erreurs

Généralités

Description des paramètres de PL7 qui dépendent de l'implémentation qui en est faite
Le tableau suivant liste en regard des tableaux de caractéristiques décrites dans le standard IEC 1131-3, l'implémentation faite dans les langages PL7.

Tableau de caractéristiques

Situations d'erreurs:

| Situations d'erreurs | Limitation et comportement de PL7 |
|---|--|
| Erreurs de conversion de type | Signalé lors de l'exécution en positionnant un bit système: cf table Eléments communs: tableau 22, caractéristique 1 |
| Le résultat numérique dépasse la plage pour le type de donnée | Signalé lors de l'exécution en le type de donnée positionnant le bit système %S18 |
| Position de caractères spécifiée invalide | Signalé lors de l'exécution en positionnant le bit système %S18 |
| Le résultat dépasse la longueur maximale de la chaîne | Signalé lors de l'exécution en de la chaîne positionnant le bit système %S15 |
| Effets de bord pendant l'évaluation d'une transition | Déecté lors de la programmation |
| Délais d'exécution non respectés | Signalé à l'exécution en positionnant le bit système %S19 |
| Autres conflits d'ordonnancement de tâches | Déecté lors de la configuration |
| Division par zéro | Déecté lors de la programmation si possible, sinon signalé |
| Type de donnée invalide pour une opération | lors de l'exécution en positionnant le bit système %S18 |
| Echec d'une itération FOR ou WHILE à se terminer | L'automate passe en défaut de débordement de chien de garde et l'unité de programmation concernée est indiquée |

Serveur OLE Automation

5

Présentation

Contenu de ce chapitre

Ce chapitre décrit le fonctionnement du serveur OLE Automation

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

| Sous-chapitre | Sujet | Page |
|---------------|----------------|------|
| 5.1 | Présentation | 72 |
| 5.2 | Mise en oeuvre | 76 |
| 5.3 | Fonctions OLE | 85 |

5.1 Présentation

Présentation

Objet de ce sous-chapitre Ce sous-chapitre décrit les généralités du serveur OLE Automation

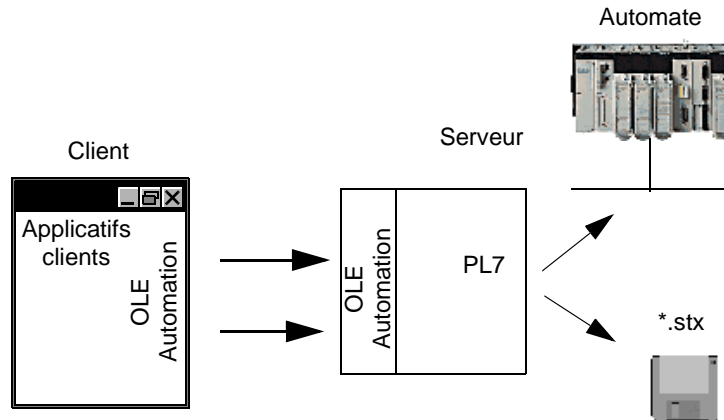
Contenu de ce sous-chapitre Ce sous-chapitre contient les sujets suivants :

| Sujet | Page |
|---|------|
| Présentation du Serveur OLE Automation | 73 |
| Modes de marche du serveur OLE Automation | 75 |

Présentation du Serveur OLE Automation

Introduction

La fonctionnalité PL7 du **Serveur OLE Automation** offre la possibilité de "piloter" l'exécution d'un PL7 de manière programmée, non interactive (sans intervention de l'opérateur) pour des applications externes à PL7. On utilise la terminologie d'applications clientes du PL7 serveur OLE Automation.



Suivant totalement le standard défini par Microsoft dans son offre OLE pour répondre à ce besoin PL7 dispose désormais d'un interface standard, public, répandu, propre et indépendant du langage de programmation de l'application cliente.

Les principales caractéristiques sont :

- l'automatisation de tâches répétitives
- la délégation à PL7 de traitement que lui seul sait exécuter
- l'accès à des informations sur les applications contenues dans les fichiers *.stx
- plus généralement le développement de fonction de niveau supérieur à PL7, de type intégrateur, tout en encapsulant les détails d'implémentation spécifique à PL7.

**Contexte
d'exécution**

La fonctionnalité PL7 serveur OLE Automation est intégrée à PL7 Pro. Elle s'installe et s'exécute avec les mêmes conditions d'installation et d'exécution que PL7. PL7 serveur OLE Automation s'appuie sur le run-time OLE Automation Microsoft fourni avec Windows 95, 98 et NT

Note : La réalisation d'un applicatif client du serveur OLE nécessite de connaître l'un des langages suivants pour la programmation OLE Automation :

- Microsoft Visual Basic, version 5.0 ou supérieure
 - Microsoft Visual C++, version 4.2 ou supérieure
 - Microsoft VBA dans Excel, version 5.0 ou supérieure.
-

Modes de marche du serveur OLE Automation

Généralités

Au mode de marche de PL7 connu jusqu'à présent, que l'on désigne comme "interactif", où le PL7 réagit uniquement aux entrées de l'opérateur, s'ajoute le mode de marche serveur OLE automation dans lequel PL7 réagit en plus à des commandes émises par une application cliente OLE.

Description

Le choix du mode de marche est fonction de la procédure de lancement de PL7:

- Le **mode interactif** est initialisé au lancement de PL7 via Windows (menu Démarrer / Programmes/ Modicon Telemecanique/PL7 Pro.).
- Le **mode serveur** est choisi si on invoque le serveur PL7 OLE Automation dans la programmation d'une application cliente OLE. Une instance ne peut pas changer de mode en cours.

A chaque fois qu'une application cliente invoque le PL7 serveur OLE Automation, une instance spécifique de PL7 est lancée, indépendamment d'autres éventuelles instances déjà lancées. Un nombre quelconque d'instances de PL7 en mode serveur ou en mode interactif peuvent cohabiter en exécution sur le même poste. Il y a parfaite indépendance des instances entre elles qui peuvent chacune s'exécuter dans leur propre contexte.

Pour toutes ces instances, les mêmes règles d'exclusion d'accès concurrent à une application STX ou à un automate s'appliquent à savoir: une application STX ou un automate ne peut être manipulé que par une instance à la fois.

Cette règle a été aménagée pour PL7 serveur OLE Automation qui peut ouvrir une application STX déjà ouverte mais dont l'enregistrement sera interdit. De même, il peut effectuer, sur demande explicite, un transfert automate —> PC depuis un automate réservé par ailleurs.

Une application cliente peut instancier plusieurs "PL7 serveur OLE Automation" concurrents.

5.2 Mise en oeuvre

Présentation

Objet de ce sous-chapitre Ce sous-chapitre décrit la mise en oeuvre du serveur OLE Automation

Contenu de ce sous-chapitre Ce sous-chapitre contient les sujets suivants :

| Sujet | Page |
|---|------|
| Installation de OLE Automation | 77 |
| Mécanisme d'accès au serveur PL7 OLE Automation | 78 |
| Lancement du serveur en mode local (mode COM) | 79 |
| Lancement du serveur en mode distant (DCOM) | 80 |
| Mise en oeuvre du serveur en mode distant | 81 |
| Modes d'exécution PL7 serveur | 82 |
| Points d'entrée : Fonction OLE | 83 |

Installation de OLE Automation

Description

L'installation du produit PL7 Serveur OLE Automation est transparente pour l'utilisateur, elle est assurée implicitement lors de l'installation de PL7.

L'installation comprend :

- Quatre exemples de client OLE Automation, complets avec leur code source et un fichier README sont fournis dans le logiciel.

Ces exemples correspondent à :

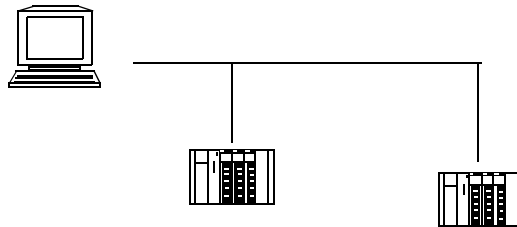
- un client visual C++ 4.2 "minimal" : le minimum à connaître pour écrire un client OLE.
 - un client visual C++ 4.2 "complet" qui exploite tous les points d'entrée disponibles.
 - un client visual Basic 5.0 qui exploite tous les points d'entrée disponibles.
 - un client Excel.
 - Un fichier *.h qui définit les valeurs des codes d'erreur rendus par le serveur.
 - Un fichier TLB d'interface pour un client Visual C++.
-

Mécanisme d'accès au serveur PL7 OLE Automation

Le PL7 serveur OLE Automation permet deux modes d'accès aux services qu'il fournit.

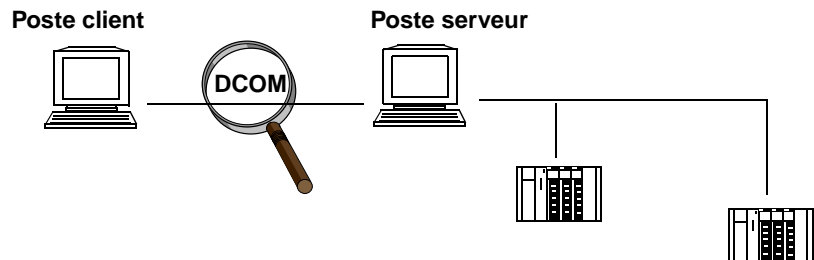
Accès en local

L'appliquatif client et PL7 serveur OLE Automation sont sur le même poste.



Accès en distant par l'intermédiaire de DCOM

L'application client et le serveur PL7 Automation sont sur des postes distincts, reliés par le réseau TCP/IP :



Lancement du serveur en mode local (mode COM)

Introduction

Le mode COM (Component object Model) définit une interface de communication entre applications. Elle permet d'être indépendante vis à vis des outils de développement.

Lancement du serveur en mode local (mode COM)

Pour lancer le serveur en mode local, suivre la procédure suivante:

| | |
|---|---|
| 1 | déclarer un objet " A " sur l'application cliente pour communiquer avec l'application serveur |
| 2 | Exécuter la fonction Createdispatch sur l'objet " A " de l'application cliente afin d'instancier l'application serveur |
| 3 | Enregistrer PL7Pro dans la base de registre comme serveur OLE, pour un fonctionnement correct en mode serveur (le lien entre l'application cliente et l'application serveur est créé si ce dernier est référencé dans la base de registre). |
| 4 | L'application cliente peut maintenant interroger l'application serveur via l'interface accessible de l'objet " A ". Ex.: OpenStx("C:\appli.stx). |

Lancement du serveur en mode distant (DCOM)

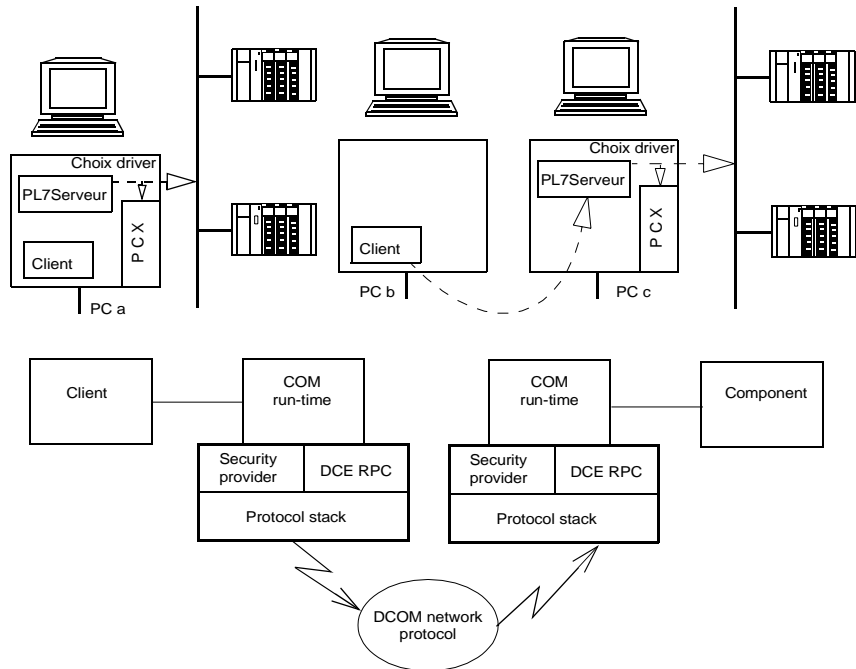
Introduction

Le PL7 serveur s'exécute en fonction des informations présentes dans la base de registre. Il peut être localisé sur la machine cliente ou sur une machine distante.

Le mode DCOM est en fait une extension du mode COM (Component Object Model). Le mode COM permet de dialoguer entre une application cliente et une application serveur sur une même machine.

Le mode DCOM permet de dialoguer entre deux machines distantes. Il remplace les protocoles de communication inter-process par les protocoles réseau.

Illustration du fonctionnement de OLE Automation en mode serveur distant:



Utilitaire de configuration

Pour établir la communication entre deux machines, nous utiliserons un utilitaire de configuration nommé " DCOMCNFG.EXE ".

Par défaut l'utilitaire " DCOMCNFG.EXE " n'est pas présent sous Windows 95. L'utilisateur doit se connecter sur internet à l'adresse suivante: " <http://www.microsoft.com> " et faire une recherche depuis ce site pour télécharger cet utilitaire. Ceci afin de faire fonctionner le serveur en DCOM sous Windows 95.

Mise en oeuvre du serveur en mode distant

Exemple d'utilisation

Soit une machine A (Windows NT 4) installée avec PL7Pro serveur.
L'utilisateur des machines doit se connecter sous un compte " administrateur " afin de pouvoir gérer les **droits d'accès**.
Le client est la machine B (Windows 95).

Procédure

Suivre la démarche ci-dessous:

| | |
|---|---|
| 1 | <p>Sur la machine B, lancer DCOMCNFG:</p> <ul style="list-style-type: none"> ● Sélectionner l'onglet «Default Properties» afin de visualiser les informations suivantes : <ul style="list-style-type: none"> ● [X] "Enable Distributed COM on this computer" ● Default Authentication Level: CONNECT ● Default Impersonation Level: Identify ● Sur l'onglet «Default Security», cocher l'option : <ul style="list-style-type: none"> ● [X] Enable remote connection ● Sur l'onglet «Application» : <ul style="list-style-type: none"> ● Sélectionner le serveur PL7Pro " PL7 server ", puis " properties " |
| 2 | <p>Sur la machine A, lancer DCOMCNFG:</p> <ul style="list-style-type: none"> ● Sélectionner l'onglet «Default Properties» afin de visualiser les informations suivantes : <ul style="list-style-type: none"> ● [X] «Enable Distributed COM on this computer ● Default Authentication Level: CONNECT ● Default Impersonation Level: Identify |
| 3 | La machine B "Client" peut piloter " PL7Pro serveur " localisé sur la machine A. |
| 4 | Sélectionner l'onglet «Default security», ajouter les utilisateurs qui auront les droits en écriture sur la machine. |

Pour en savoir plus

Il existe des Groupes de discussion ("newsgroups") sur Internet, où il est possible de poser des questions concernant DCOM.

Modes d'exécution PL7 serveur

Introduction

Le PL7 serveur OLE Automation possède deux modes d'exécution que l'on peut choisir dynamiquement en cours d'exécution (voir `SetIHMServer`)

Modes d'exécution

- Mode d'exécution sans IHM. Le PL7 est lancé "en tâche de fond", sans aucun affichage ni possibilité d'entrée de l'opérateur. C'est le mode typique à utiliser par exemple pour automatiser les tâches répétitives ou accéder à des informations sur une application contenue dans un fichier stx.
- Mode d'exécution avec IHM. Le PL7 est lancé comme un PL7 "interactif" avec affichage et possibilité d'entrée de l'opérateur mais il reste réceptif aux commandes émises par son application cliente. Ce mode a été développé pour visualiser des éléments de programme ou de configuration d'une application contenue dans un fichier stx depuis des outils externes comme DIAG Viewer.

Note : Les services concernant la visualisation de programmes, d'outils, de modules ne s'exécutent que dans le mode avec IHM.

Le PL7 serveur OLE Automation dans son mode avec IHM est soumis aux contrôle droits utilisateur. Il est positionné en profil droit utilisateur "Read Only", qui correspond à son rôle d'afficheur de code et modules E/S PL7.

Le PL7 serveur OLE Automation en mode sans IHM n'est pas soumis aux contrôle droits utilisateur mais les services offerts ne permettent pas de modifier l'application.

Points d'entrée : Fonction OLE

Introduction

Les points d'entrée sont organisés en quatre types:

- **Contexte d'exécution**
- **Contrôle API**
- **Lecture information**
- **Visualisation d'élément de l'application**

Contexte d'exécution

Points d'entrée

| Nom | Description |
|----------------------|--|
| OpenStx | Ouverture d'une application |
| SaveStx | Enregistrement de l'application active |
| CloseStx | Fermeture de l'application active |
| Set_DriverAndAddress | Modification adresse et driver de l'automate accédé |
| SetServerIHM | Rend le serveur PL7 OLE Automation interactif ou non |
| GetPL7IHM | Donne l'état: application ouverte ou fermée, mode local - connecté, état API |
| GetMessageError | Lit le message d'erreur associé au code erreur |

Contrôle API

Points d'entrée

| Nom | Description |
|-----------------|--|
| ConnectPLC | Entre dans le mode connecté |
| DisconnectPLC | Sort du mode connecté |
| SenCommandToPLC | Lance une commande à l'automate (RUN, STOP INIT) |
| DownloadToPLC | Chargement de l'application active dans un automate |
| UploadFromPLCM | Recopie d'une application automate dans l'application active |

**Lecture
information**

Points d'entrée

| Nom | Description |
|-------------------|---|
| ExportScyFile | Export des symboles dans l'application active sous forme de fichier scy |
| ExportFefFile | Export de l'application active sous forme de fichier de type fef |
| GetSymbol | Lit le symbole et le commentaire associé à un repère |
| GetSTXAppIdentity | Lit les informations générales d'une application contenue dans un fichier STX |
| GetPLCAppIdentity | Lit les informations générales d'une application contenue dans un automate |
| GetServerVersion | Lit la version du serveur |

**Visualisation
d'élément de
l'application**

Points d'entrée

| Nom | Description |
|-----------------|---|
| SetPosPL7Window | Modifie les caractéristiques d'affichage du PL7 (position et forme) |
| ShowProgram | Ouverture d'un éditeur sur un module programme donné |
| CloseProgram | Fermeture d'un éditeur programme |
| ShowIOModule | Ouverture d'un éditeur sur un module d'E/S donné |
| CloseIOModule | Fermeture d'un éditeur de module d'E/S/S donné |
| ShowDFB | Ouverture d'un éditeur sur le code d'un DFB donné |
| CloseDFB | Fermeture d'un éditeur sur le code d'un DFB |
| OpenTool | Ouverture d'un outil MDI quelconque sans contexte |

Note : Les points d'entrée concernant la visualisation d'élément de l'application ne fonctionnent pas si le serveur est mode sans IHM.

5.3 Fonctions OLE

Présentation

Objet de ce sous-chapitre Ce sous-chapitre décrit les fonctions OLE du serveur OLE Automation

Contenu de ce sous-chapitre Ce sous-chapitre contient les sujets suivants :

| Sujet | Page |
|--------------------------------|------|
| OpenStx | 87 |
| CloseStx | 88 |
| OpenStxWithMaj | 89 |
| ExportScyFile | 90 |
| ExportFeFile | 91 |
| ExportFefFileWithConfiguration | 92 |
| DisconnectPLC | 93 |
| ConnectPLC | 94 |
| SaveStx | 95 |
| DownloadDataToPLC | 96 |
| DownloadToPLC | 98 |
| UploadFromPLC | 99 |
| UploadDataFromPLC | 100 |
| UploadFromPLCWithSymbol | 102 |
| GetSymbol | 104 |
| SetServerIHM | 105 |
| SetDownloadFunction | 106 |
| GetPL7State | 107 |
| GetSTXApplIdentity | 108 |
| GetPLCApplIdentity | 109 |
| SendCommandToPLC | 111 |
| SetDriverAndAdresse | 112 |
| OpenTool | 113 |
| SetPosPL7Windows | 115 |
| ShowProgram | 116 |
| CloseProgram | 117 |

| Sujet | Page |
|------------------|-------------|
| ShowIOModule | 118 |
| CloseIOModule | 119 |
| ShowDFB | 120 |
| CloseDFB | 121 |
| GetMessageError | 122 |
| GetServerVersion | 123 |
| ImportScyFile | 124 |

OpenStx

Généralités

Cette fonction permet d'ouvrir un fichier de type application stx.

Description

Syntaxe :

```
integer OpenStx(String lpAppPathName)
```

- **Entrée :**
Saisir la chaîne de caractères contenant le nom du fichier que l'utilisateur désire ouvrir.
- **Fonction appelée :**
`OpenStation` : Fonction de la gestion station (`gesta.dll`).
- **Contexte nominal :**
Le serveur PL7 est actif.
- **Cas d'erreur :**

| | |
|-----------------------------|---|
| SRV_ERR_GEN_FILENOTFOUND | Le fichier n'a pas été trouvé. <code>OpenStation</code> |
| SRV_ERR_OPEN_BADZIP | Problème de décompression fichier |
| SRV_ERR_OPEN_BADFILE | Problème de lecture du fichier stx |
| SRV_ERR_OPEN_NOK_COMPATIBLE | Le processeur est incompatible avec PL7 ouvert. |
| SRV_ERR_OPEN_OPEN | Une application est déjà ouverte |
| SRV_ERR_GEN_PARAM_EMPTY | Le Path Name est vide |
| SRV_ERR_GEN_ACTION | Erreur lors de l'ouverture de l'application. |
| SRV_ERR_GEN_DRIVE FULL | Plus de place libre sur le disque pour l'ouverture de l'application |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'une ouverture réussit, soit un code erreur.

CloseStx

Généralités

Cette fonction permet de fermer l'application courante

Description

Syntaxe :

```
integer CloseStx( integer p_bWithoutSave)
```

- **Entrée :**
Si l'application a été modifiée, il est possible de la fermer sans avertir l'utilisateur (`p_bWithoutSave` a `TRUE`)
- **Fonction appelée :**
`CloseStation` : Fonction de la gestion station (`gesta.dll`).
- **Contexte nominal :**
Le serveur PL7 est actif, une application est ouverte.
- **Cas d'erreur :**

| | |
|-----------------------|---|
| SRV_ERR_GEN_NOTOPEN | Il n'y a pas d'application ouverte. |
| SRV_ERR_CLOSE_NOTSAVE | L'application a été modifiée, il faut l'enregistrer afin de sortir. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'une fermeture réussie, soit un code erreur.
-

OpenStxWithMaj

Généralités Cette fonction permet d'exporter l'application active avec sa configuration dans un fichier de type fef.

Description

Syntaxe :

integer **OpenStxWithMaj** (string NameFile, integer WithNewCat)

- **Entrée :**
 - Saisir 1 chaînes de caractères contenant le nom et le chemin du fichier stx que l'on veut ouvrir.
 - Un entier nous indiquant si nécessaire si il faut mettre à jour les fichiers catalogue ou non (valeur 0 pour pas de mise à jour ou 1 pour une mise à jour).
- **Fonction appelée :**
OpenStation : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
 - Le serveur PL7 est actif,
 - PL7 est en mode local.
- **Cas d'erreur :**

| | |
|--------------------------|--|
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_PARAM_EMPTY | Le nom du fichier dat est incorrect ou vide. |
| SRV_ERR_GEN_DRIVEFULL | Plus de place libre sur le disque pour l'ouverture de l'application. |
| SRV_ERR_OPEN_OPEN | Une application est déjà ouverte. |
| SRV_ERR_GEN_FILENOTFOUND | Le fichier n'a pas été trouvé. |
| SRV_ERR_OPEN_BADZIP | Problème d'extraction de fichier zip. |

- **Sortie :**
Code retour de type short. Soit 0 dans le cas d'une ouverture de station réussie, soit un code d'erreur.
- **Limitations :**
Le serveur de PL7, sans IHM, une fois l'importation réussie, ne peut pas fonctionner en mode avec IHM sans une sauvegarde préalable de l'application ouverte.

ExportScyFile

Généralités

Cette fonction permet d'exporter les symboles contenus dans l'application active sous forme de fichier de type scy.

Description

Syntaxe :

| |
|---|
| integer ExportScyFile (String p_psScyFile) |
|---|

- **Entrée :**
Saisir la chaîne de caractères contenant le nom du fichier scy.
- **Fonction appelée :**
`ExportScyFile` : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif. Une application est ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_ACTION | Erreur lors de la création du fichier sourcesymbole. |
| SRV_ERR_GEN_NOTOPEN | Il n'y a pas d'application ouverte. |
| SRV_ERR_GEN_PARAM_EMPTY | <code>p_ psScyFile</code> est vide |
| SRV_EXPORT_ERRFILE | Erreur d'écriture du fichier source. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un export réussi, soit un code erreur.
-

ExportFeFile

Généralités Cette fonction permet d'exporter l'application active sous forme de fichier de type fef.

Description

Syntaxe :

```
integer ExportFefFile(String p_psNamefile)
```

- **Entrée :**
Saisir la chaîne de caractères contenant le nom du fichier fef .
- **Fonction appelée :**
`ExportFefFile` : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif. Une application est ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|---|
| SRV_ERR_GEN_ACTION | Erreur lors de la création du fichier source application. |
| SRV_ERR_GEN_PARAM_EMPTY | Il n'y a pas d'application ouverte. |
| SRV_ERR_GEN_PARAM_EMPTY | <code>p_ psNamefile</code> est vide. |
| SRV_EXPORT_ERRFILE | Erreur d'écriture du fichier source. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un export réussi, soit un code erreur.

ExportFefFileWithConfiguration

Généralités

Cette fonction permet d'exporter l'application active avec sa configuration dans un fichier de type fef.

Description

Syntaxe :

| |
|--|
| integer ExportFefFileWithConfiguration (string FefFile) |
|--|

- **Entrée :**
Saisir 1 chaînes de caractères contenant le nom et le chemin du fichier scy que l'on va exporter.
- **Fonction appelée :**
ExportFefFileWithConf : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
 - Le serveur PL7 est actif,
 - PL7 est en mode local, station ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_PARAM_EMPTY | Le nom du fichier dat est incorrect ou vide. |
| SRV_ERR_GEN_NOTOPEN | Pas d'application STX ouverte. |
| SRV_EXPORT_ERRFILE | Impossible de créer le fichier à exporter. |

- **Sortie :**
Code retour de type short. Soit 0 dans le cas d'une exportation réussie, soit un code d'erreur.
-

DisconnectPLC

Généralités Cette fonction permet de réaliser la déconnexion entre l'automate et PL7.

Description **Syntaxe :**

```
integer ExportFefFile(String p_psNamefile)
```

- **Fonction appelée :**
DisconnectStation : Fonction de la gestion station (gesta.dll).
 - **Contexte nominal :**
Le serveur PL7 est actif.
 - **Sortie :**
Code retour de type short. Soit 0 dans le cas d'une déconnexion réussie, soit un code erreur (SRV_ERR_GEN_ACTION).
-

ConnectPLC

Généralités

Cette fonction permet de se connecter à un automate.

Description

Syntaxe :

```
integer ConnectPLC(String p_lpDriver, String p_lpAddress)
```

- **Entrée :**
Saisir deux chaînes de caractères contenant le driver et l'adresse de l'automate.
- **Fonction appelée :**
`ConnectStation` : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif. Dans le cas où les paramètres sont des chaînes vides, le serveur exécute la fonction avec le driver et/ou l'adresse courant.
- **Cas d'erreur :**

| | |
|----------------------------|---|
| SRV_ERR_GEN_CONNECT | La connexion n'a pas pu s'effectuer. |
| SRV_ERR_GEN_COMMUNICATION | Problème de communication entre PL7pro et l'automate |
| SRV_ERR_GEN_RESERVED | L'automate est réservé il est donc impossible de s'y connecter. |
| SRV_ERR_PL7_BLANK | L'automate ne contient pas d'application, il est donc impossible de se connecter. |
| SRV_ERR_PC_BLANK | Il n'y a pas d'application ouverte, il est donc impossible de se connecter. |
| SRV_ERR_DIFFERENCE_PL7_STX | L'application ouverte et l'application contenue dans l'automate sont différentes, il est donc impossible de se connecter. |
| SRV_ERR_GEN_ADDRESS | Le paramètre adresse est incorrect. |
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_DRIVER | Le paramètre driver est incorrect. |
| SRV_ERR_GEN_NOTOPEN | Il n'y a pas d'application ouverte. |
| SRV_ERR_GEN_PROTECTEDAPPLI | Application protégée. |
| SRV_COMPATIBLE_PL7 | Problème de compatibilité avec l'automate. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'une connexion réussie, soit un code erreur.
-

SaveStx

Généralités

Cette fonction permet de sauvegarder l'application ouverte.

Description

Syntaxe :

```
integer SaveStx(String p_lpStxFile)
```

- **Entrée :**
Saisir une chaîne de caractères contenant le path et le nom du fichier de sauvegarde.
- **Fonction appelée :**
`SaveStx` : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif. Une application est ouverte, PL7 est en local.
- **Cas d'erreur :**

| | |
|-------------------------|---|
| SRV_ERR_SAVEDENIED | Accès au fichier refusé |
| SRV_ERR_SAVEERRZIP | Erreur lors du compactage de l'application |
| SRV_ERR_SAVEERRREN | Renommage refusé |
| SRV_ERR_GEN_NOTOPEN | Il n'y a pas d'application ouverte |
| SRV_ERR_GEN_PARAM_EMPTY | <code>p_ lpStxFile</code> est vide. |
| SRV_ERR_GEN_ACTION | Erreur lors de la sauvegarde de l'application. |
| SRV_ERR_PL7_CONNECT | La sauvegarde ne peut s'effectuer qu'en mode local. |

- **Sortie :**
Code retour de type short. Soit 0 dans le cas d'une sauvegarde réussie, soit un code erreur.

DownloadDataToPLC

Généralités

Cette fonction permet de charger un fichier contenant des mots dans l'automate.

Description

Syntaxe :

integer **DownLoadDataToPLC** (string Driver, string Address, string NameFile)

- **Entrée :**
Saisir les 3 chaînes de caractères contenant le driver, l'adresse de l'automate et le nom complet du fichier au format dat.
- **Fonction appelée :**
DownloadData : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
 - Le serveur PL7 est actif,
 - PL7 est en mode local,
 - Si les paramètres (driver et adresse) sont des chaînes vides, le serveur exécute la fonction avec le driveur et/ou l'adresse courante.
- **Cas d'erreur :**

| | |
|---------------------------|--|
| SRV_ERR_DOWNLOAD_CART | Problème de compatibilité de cartouche entre l'application ouverte et l'automate. |
| SRV_ERR_COMPATIBLE_PLC | Problème de compatibilité entre l'application ouverte et l'automate. |
| SRV_ERR_PC_BLANK | Il n'y a pas d'application ouverte. |
| SRV_ERR_GEN_ADDRESS | Le paramètre <code>p_IpAddress</code> est incorrect. |
| SRV_ERR_GEN_DRIVER | Le paramètre <code>p_IpDriver</code> est incorrect. |
| SRV_ERR_GEN_RESERVED | L'automate est réservé, il est donc impossible de s'y connecter. |
| SRV_ERR_GEN_COMMUNICATION | Problème de communication entre PL7pro et l'automate. |
| SRV_ERR_PLC_CONNECT | Il faut être en mode local pour pouvoir effectuer le download. |
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_PARAM_EMPTY | Le nom du fichier dat est incorrect ou vide. |
| SRV_ERR_GEN_PROTECTEAPPLI | L'application est protégée en lecture, il n'est pas possible de se connecter à l'automate. |
| SRV_ERR_TRFDATA_READDS | Erreur de communication. |

| | |
|----------------------------|--|
| SRV_ERR_TRFDATA_NODATA | Variables numériques non configurées ou incorrectement lues. Le fichier n'est pas créé. |
| SRV_ERR_TRFDATA_READWRITE | Problème sur le fichier dat. |
| SRV_ERR_TRFDATA_INFOFILE | Problème sur le fichier dat. |
| SRV_ERR_TRFDATA_RUNREFUSED | L'automate a refusé la demande de passage en RUN (Entrée RUN/STOP positionnée sur STOP). |
| SRV_ERR_TRFDATA_READOBJ | Erreur de lecture ou d'écriture des variables numériques. |
| SRV_ERR_TRFDATA_WRITEOBJ | Erreur de lecture ou d'écriture des variables numériques. |
| SRV_TRFDATA_OVERFLOW | Attention, la plage de valeurs était trop grande, il y a eu débordement. |
| SRV_TRFDATA_BORNENOCORRECT | Les valeurs minimum et maximum sont incorrectes. |

- **Sortie :**

Code retour de type short. Soit 0 dans le cas d'un chargement réussi, soit un code d'erreur.

DownloadToPLC

Généralités

Cette fonction permet de télécharger une application en mémoire automate.

Description

Syntaxe :

```
integer DownloadToPLC(String p_lpDriver, String p_lpAdresse)
```

- **Entrée :**
Saisir deux chaînes de caractères contenant le driver et l'adresse de l'automate.
- **Fonction appelée :**
DownloadStation: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif. Une application est ouverte, PL7 est en local. Dans le cas où les paramètres sont des chaînes vides, le serveur exécute la fonction avec le driver et/ou l'adresse courant.
- **Cas d'erreur :**

| | |
|---------------------------|--|
| SRV_ERR_DOWNLOAD_CART | Problème de compatibilité de cartouche mémoire entre l'application ouverte et l'automate |
| SRV_ERR_COMPATIBLE_PL7 | Problème de compatibilité entre l'application ouverte et l'automate. |
| SRV_ERR_GEN_ADDRESS | Le paramètre <code>p_lpAdresse</code> est incorrect. |
| SRV_ERR_GEN_DRIVER | Le paramètre <code>p_lpDriver</code> est incorrect. |
| SRV_ERR_GEN_NOTOPEN | Il n'y a pas d'application ouverte. |
| SRV_ERR_GEN_RESERVED | L'automate est réservé. |
| SRV_ERR_GEN_COMMUNICATION | Problème de communication. |
| SRV_ERR_PL7_CONNECT | L'automate est déjà connecté |
| SRV_ERR_PL7_ACTION | Erreur lors du dowload.. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
-

UploadFromPLC

Généralités Cette fonction permet de recopier en mémoire une application contenue dans un automate.

Description

Syntaxe :

```
integer UploadFromPLC(String p_lpDriver, String p_lpAdresse, integer p_iReservedMode)
```

- **Entrée :**
Saisir deux chaînes de caractères contenant le driver et l'adresse de l'automate. Un integer (`p_iReservedMode`) permettant de faire un Upload sur un automate réservé.
- **Fonction appelée :**
`UploadStation`: Fonction de la gestion station (`gesta.dll`).
- **Contexte nominal :**
Le serveur PL7 est actif. Pas d'application ouverte, et PL7 est en mode local. Si les paramètres sont des chaînes vides, le serveur exécute la fonction avec le driver et/ou l'adresse courant.
- **Cas d'erreur :**

| | |
|---------------------------|--|
| SRV_ERR_UPLOAD_TRANSFER | Problème lors du transfert de l'application contenue dans l'automate |
| SRV_ERR_UPLOAD_ABORT | Suspension du transfert. |
| SRV_ERR_UPLOAD_RESERVED | Application réservée. |
| SRV_ERR_UPLOAD_NOAPPLI | Automate vierge. |
| SRV_ERR_GEN_ADDRESS | Adresse incorrecte. |
| SRV_ERR_GEN_ACTION | Erreur d'exécution de la fonction. |
| SRV_ERR_PLC_CONNECT | Erreur de connexion. |
| SRV_ERR_GEN_DRIVER | Driver incorrect. |
| SRV_ERR_GEN_COMMUNICATION | Erreur de communication. |
| SRV_ERR_PLC_CONNECT | L'automate est en mode connecté. |
| SRV_OPEN_NOT_COMPATIBLE | Problème de compatibilité. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.

UploadDataFromPLC

Généralités

Cette fonction permet de télécharger un fichier contenant des mots contenus dans l'automate.

Description

Syntaxe :

integer **UpLoadDataFromPLC** (string Driver, string Address, integer ValMin, integer ValMax, string NameFile)

- **Entrée :**
 - Saisir les 3 chaînes de caractères contenant le driver, l'adresse de l'automate et le nom complet du fichier au format dat,
 - Saisir les 2 entiers qui sont les valeurs minimum et maximum que l'on désire récupérer.
- **Fonction appelée :**
UploadData : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
 - Le serveur PL7 est actif,
 - PL7 est en mode local,
 - Si les paramètres (driver et adresse) sont des chaînes vides, le serveur exécute la fonction avec le driver et/ou l'adresse courante.
- **Cas d'erreur :**

| | |
|---------------------------|--|
| SRV_ERR_COMPATIBLE_PLC | Problème de compatibilité entre l'application ouverte et l'automate. |
| SRV_ERR_GEN_ADDRESS | Le paramètre <code>p_IpAddress</code> est incorrect. |
| SRV_ERR_GEN_DRIVER | Le paramètre <code>p_IpDriver</code> est incorrect. |
| SRV_ERR_GEN_RESERVED | L'automate est réservé, il est donc impossible de s'y connecter. |
| SRV_ERR_GEN_COMMUNICATION | Problème de communication entre PL7pro et l'automate. |
| SRV_ERR_PLC_CONNECT | Il faut être en mode local pour pouvoir effectuer le download. |
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_PARAM_EMPTY | Le nom du fichier dat est incorrect ou vide. |
| SRV_ERR_GEN_PROTECTEAPPLI | L'application est protégée en lecture, il n'est pas possible de se connecter à l'automate. |
| SRV_ERR_TRFDATA_READDS | Erreur de communication. |

| | |
|----------------------------|--|
| SRV_ERR_TRFDATA_NODATA | Variables numériques non configurées ou incorrectement lues. Le fichier n'est pas créé. |
| SRV_ERR_TRFDATA_READWRITE | Problème sur le fichier dat. |
| SRV_ERR_TRFDATA_INFOFILE | Problème sur le fichier dat. |
| SRV_ERR_TRFDATA_RUNREFUSED | L'automate a refusé la demande de passage en RUN (Entrée RUN/STOP positionnée sur STOP). |
| SRV_ERR_TRFDATA_READOBJ | Erreur de lecture ou d'écriture des variables numériques. |
| SRV_ERR_TRFDATA_WRITEOBJ | Erreur de lecture ou d'écriture des variables numériques. |
| SRV_TRFDATA_OVERFLOW | Attention, la plage de valeurs était trop grande, il y a eu débordement. |
| SRV_TRFDATA_BORNENOCORRECT | Les valeurs minimum et maximum sont incorrectes. |

- **Sortie :**

Code retour de type short. Soit 0 dans le cas d'un déchargement réussi, soit un code d'erreur.

UploadFromPLCWithSymbol

Généralités

Cette fonction permet de télécharger une application avec les symboles associés contenus dans l'automate.

Description

Syntaxe :

integer **UpLoadFromPLCWithSymbol** (string Driver, string Address, integer ReserveMode)

- **Entrée :**
Saisir les 2 chaînes de caractères contenant le driver, l'adresse de l'automate.
Un entier (`p_IReserveMode`) permettant de faire un transfert PLC vers PC sur un automate réservé.
- **Fonction appelée :**
`UploadStationWithSymbol` : Fonction de la gestion station (`gesta.dll`).
- **Contexte nominal :**
 - Le serveur PL7 est actif,
 - PL7 est en mode local,
 - Si les paramètres (driver et adresse) sont des chaînes vides, le serveur exécute la fonction avec le driver et/ou l'adresse courante.
- **Cas d'erreur :**

| | |
|---------------------------|--|
| SRV_ERR_COMPATIBLE_PL7 | Problème de compatibilité entre l'application ouverte et l'automate. |
| SRV_ERR_GEN_ADDRESS | Le paramètre <code>p_IpAddress</code> est incorrect. |
| SRV_ERR_GEN_DRIVER | Le paramètre <code>p_IpDriver</code> est incorrect. |
| SRV_ERR_GEN_RESERVED | L'automate est réservé, il est donc impossible de s'y connecter. |
| SRV_ERR_GEN_COMMUNICATION | Problème de communication entre PL7pro et l'automate. |
| SRV_ERR_PL7_CONNECT | Il faut être en mode local pour pouvoir effectuer le download. |
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_PARAM_EMPTY | Le nom du fichier <code>dat</code> est incorrect ou vide. |
| SRV_ERR_GEN_PROTECTEAPPLI | L'application est protégée en lecture, il n'est pas possible de se connecter à l'automate. |
| SRV_OPEN_NOK_COMPATIBLES | Le serveur ne supporte pas le niveau d'application STX à ouvrir. |
| SRV_ERR_UPLOAD_NOAPPLI | Une erreur est survenue pendant le transfert d'application. |

| | |
|------------------------------------|---|
| SRV_ERR_UPLOAD_READINFOSYMB | Erreur de communication lors de la lecture des informations embarqués de la base.. |
| SRV_ERR_UPLOAD_WRITEINFOSYMBOL | Erreur d'écriture. Si le disque de sauvegarde est saturé sur le fichier dat. |
| SRV_ERR_UPLOAD_INVALIDSYMB | Base de symboles embarqués dans l'automate invalide. |
| SRV_ERR_UPLOAD_INVALIDVERSION | La version de la base des symboles embarquée n'est pas supportée par cette version de PL7. |
| SRV_ERR_UPLOAD_SYMBERR | Erreur interne sur la base des symboles embarqués. |
| SRV_ERR_UPLOAD_NBSYMBERR_NOCORRECT | Incohérence sur les objets stockés dans la base compressée |
| SRV_ERR_UPLOAD_SYMBFILE | Problème de lecture de la base contenue dans l'automate. |
| SRV_ERR_UPLOAD_READSYMBFILE | Erreur de lecture de la base contenue dans l'automate. |
| SRV_ERR_UPLOAD_PBMEM | La mémoire est saturée, l'opération ne peut pas se dérouler normalement. |
| SRV_ERR_UPLOAD_ABORTSYMB | Abandon de la lecture du fichier de la base des symboles embarqués. |
| SRV_ERR_UPLOAD_CHECKSYMB | Une différence a été détectée entre la "checksum" embarquée et la "checksum" recalculée. |
| SRV_ERR_UPLOAD_DIFFSYMB | Une différence a été détectée entre les informations transférées et les informations recalculées. |

- **Sortie :**

Code retour de type short. Soit 0 dans le cas d'un déchargement réussi, soit un code d'erreur.

GetSymbol

Généralités

Cette fonction permet de donner le symbole et le commentaire associé à un repère.

Description

Synthèse :

| |
|--|
| integer GetSymbol (String p_lpRepere, String* p_bsSymbole, String* p_bsComment) |
|--|

- **Entrée :**
Saisir la chaîne de caractères contenant le repère à modifier ou à renseigner.
- **Fonction appelée :**
`GetSymbol`: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif. Une application est ouverte et en mode local.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_ADDRESSSYNTAX | la chaîne de caractères passée en paramètre ne correspond pas à un repère. |
| SRV_ERR_GEN_NOTOPEN | Il n'y a pas d'application ouverte. |
| SRV_ERR_GEN_PARAM_EMPTY | <code>p_lpRepere</code> est vide. |
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_ADDRESS | Adresse incorrecte. |
| SRV_ERR_GEN_ACTION | Erreur d'exécution de la fonction. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
Deux chaînes de caractères :
 - Le symbole associé (`p_bsSymbole`).
 - Le commentaire associé (`p_bsComment`).
-

SetServerIHM

Généralités

Cette fonction permet de visualiser ou pas le serveur PL7. Elle réalise le passage d'un mode de fonctionnement à l'autre. C'est à dire d'un PL7Pro serveur sans IHM à un PL7 Pro serveur avec IHM et vice et versa.

Description

Syntaxe :

| |
|--|
| integer SetServerIHM (integer p_bIHM) |
|--|

- **Entrée :**
Un booléen (affichage ou non affichage IHM).
- **Fonction appelée :**
`SetIHMServer` : Fonction de la Windows application (sawinapp.cpp).
- **Contexte nominal :**
Le serveur PL7 est actif.
- **Cas d'erreur :**

| | |
|------------------------|--|
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_MODIFAPPLI | Application en cours de modification. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.

SetDownloadFunction

Généralités

Cette fonction permet d'autoriser ou pas le transfert d'une application via le serveur dans un automate.

Description

Syntaxe :

boolean **SetDownloadFunction** (integer Download)

- **Entrée :**

Saisir 1 entier (valeur 0 ou 1) permettant d'activer ou non le transfert de l'application dans l'automate,

- **Contexte nominal :**

- Le serveur PL7 est actif,
- PL7 est en mode local.

- **Sortie :**

TRUE.

- **Limitations :**

Cette fonction n'interdit le transfert que pour une utilisation en mode serveur.

GetPL7State

Généralités

Cette fonction permet de donner l'état du serveur.

Description

Syntaxe :

```
integer GetPL7State(String* p_lpStation, String * p_lpConnection)
```

- **Fonction appelée :**
GetEtatPL7: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif.
- **Cas d'erreur :**

| | |
|--------------------|--|
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
|--------------------|--|

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
Deux chaînes de caractères informent l'utilisateur sur:
 - L'état de la station (ouverte ou fermée).
 - L'état du mode (local ou connecté). En mode connecté, l'état de l'automate run ou stop est indiqué.

GetSTXAppIdentity

Généralités

Cette fonction permet de connaître les informations générales d'une application.

Description

Syntaxe :

| |
|--|
| integer GetSTXAppIdentity (String p_lpNameStx, VARIANT FAR* p_pVarInfo) |
|--|

- **Entrée :**
Saisir la chaîne de caractères (p_lpNameStx) contenant le nom de l'application.
- **Fonction appelée :**
FicheIdentAppli: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_PARAM_EMPTY | p_ lpNameStx est vide. |
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_APPLIINFO_NOK | Information incorrecte. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.

Informations visualisées :

| | |
|----|---|
| 0 | Le nom de la station, |
| 1 | Numéro de version de l'application, |
| 2 | Date et heure de modification |
| 3 | La signature de la station, |
| 4 | La signature concernant la configuration des entrées/sorties locales, |
| 5 | La signature concernant la configuration des entrées/sorties deportées, |
| 6 | La signature concernant le code PL7, |
| 7 | La signature concernant le grafcet, |
| 8 | La signature concernant les constantes, |
| 9 | La signature concernant les symboles, |
| 10 | La signature concernant la réservation, |
| 11 | Le commentaire associé à l'application. |

GetPLCApplIdentity

Généralités

Cette fonction permet de connaître les informations générales d'une application contenue dans un automate.

Description

Syntaxe :

```
integer GetPLCApplIdentity(String p_lpDriver, String p_lpAdresse, VARIANTFAR* p_pvInfoAppli)
```

- **Entrée :**
Saisir les deux chaînes de caractères contenant le driver et l'adresse de l'automate.
- **Fonction appelée :**
FicheIdentAppliOnAutomate: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif et il est en mode local.
- **Cas d'erreur :**

| | |
|---------------------------|---|
| SRV_ERR_APPLIINFO_NOK | Information incorrecte. |
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_ADDRESS | Adresse incorrecte. |
| SRV_ERR_COMPATIBLE_PL7 | Problème de compatibilité entre API et le logiciel PL7. |
| SRV_ERR_PL7_BLANK | L'automate ne contient pas d'application. |
| SRV_ERR_GEN_ACTION | Problème lors du stockage des informations application. |
| SRV_ERR_GEN_DRIVER | Le paramètre <code>p_lpDriver</code> est incorrect. |
| SRV_ERR_GEN_COMMUNICATION | Erreur de communication. |

- **Sortie :**
Code retour de type short. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.

Informations visualisées :

| | |
|---|---|
| 0 | Le nom de la station, |
| 1 | Numéro de version de l'application, |
| 2 | Date et heure de modification |
| 3 | La signature de la station, |
| 4 | La signature concernant la configuration des entrées/sorties locales, |
| 5 | La signature concernant la configuration des entrées/sorties déportées, |

| | |
|----|---|
| 6 | La signature concernant le code PL7, |
| 7 | La signature concernant le grafcet, |
| 8 | La signature concernant les constantes, |
| 9 | La signature concernant les symboles, |
| 10 | La signature concernant la réservation, |
| 11 | Le commentaire associé à l'application. |

SendCommandToPLC

Généralités Cette fonction permet de lancer une commande dans l'automate.

Description

Syntaxe :

```
integer SendCommandToPLC(integer p_iCommand))
```

- **Entrée :**
Le type de la commande que l'utilisateur veut exécuter.
3 types possibles :
 - SRV_COMMAND_INIT : Commande d'initialisation de l'automate
 - SRV_COMMAND_STOP : Commande Stop
 - SRV_COMMAND_RUN : Commande Run
- **Fonction appelée :**
StationCommand : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif. Une application est ouverte, PL7 est en mode connecté, en STOP.
- **Cas d'erreur :**

| | |
|----------------------------|---|
| SRV_ERR_GEN_ACTION | Commande incorrecte. |
| SRV_ERR_GEN_NOTOPEN | Pas d'application ouverte. |
| SRV_ERR_COMMAND_NOTINLOCAL | Le serveur est en mode local. |
| SRV_COMMAND_ERRINIT | La fonction INIT est impossible, l'automate est en RUN. |

- **Sortie :**
Code retour de type short. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.

SetDriverAndAdresse

Généralités Cette fonction permet de changer le driver et l'adresse pour la station courante.

Description **Syntaxe :**

```
integer SetDriverAndAdresse(String p_lpDriver, String p_lpAdresse))
```

- **Entrée :**
Saisir les deux chaînes de caractères contenant le driver et l'adresse de l'automate.
- **Fonction appelée :**
DriverAndAdresse: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 est actif. Une application est ouverte et en mode local. Si les paramètres ne sont pas renseignés ou incorrects, les valeurs courantes sont inchangées.
- **Cas d'erreur :**

| | |
|---------------------|---|
| SRV_ERR_GEN_ADDRESS | l'adresse passée en paramètre est incorrecte. |
| SRV_ERR_GEN_DRIVER | le driver passé en paramètre est incorrect. |
| SRV_ERR_PLC_CONNECT | Le serveur est en mode connecté. |
| SRV_ERR_PLC_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_PARAM_EMPTY | Les paramètres sont vides. |

- **Sortie :**
Code retour de type *short*. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
-

OpenTool

Généralités

Cette fonction permet d'ouvrir les outils présents dans la chaîne de caractères passée en paramètre.

Description

Syntaxe :

```
integer OpenTool(String p_lpListTool)
```

- **Entrée :**
Chaînes de caractères contenant la liste des outils que le serveur doit ouvrir.
exemple de format : tools1;tools2;tools3
- **Fonction appelée :**
`OpenTool`: Fonction de la gestion station (`gesta.dll`).
- **Contexte nominal :**
Le serveur PL7 avec IHM est actif. Une application est ouverte.
- **Cas d'erreur :**

| | |
|--------------------------|--|
| SRV_ERR_GEN_ACTION | Impossible d'ouvrir l'outil. |
| SRV_ERR_GEN_PARAM_EMPTY | Paramètre vide. |
| SRV_ERR_GEN_NOTOPEN | Il n'y a pas d'application ouverte. |
| SRV_ERR_GEN_WITHIHM | Fonction disponible avec le serveur PL7 en mode IHM. |
| SRV_ERR_GEN_OPENEDTVDATA | L'outil "Table d'animation" ne peut être ouvert. |

- **Sortie :**
Code retour de type `short`.
Liste des numéros d'outils applicables à PL7

| Numéro | Désignation |
|--------|--|
| 5 | Ouvre l'éditeur LD |
| 6 | Ouvre l'éditeur Grafcet |
| 7 | Ouvre l'éditeur IL |
| 8 | Ouvre l'éditeur ST |
| 10 | Ouvre l'éditeur de configuration |
| 12 | Ouvre l'éditeur de variables |
| 13 | Ouvre le navigateur de références croisées |
| 14 | Crée une table d'animation |
| 15 | Ouvre l'éditeur de documentation |
| 188 | Ouvre l'outil d'importation d'application |

| Numéro | Désignation |
|---------------|--|
| 189 | Ouvre l'éditeur de texte |
| 207 | Ouvre l'écran d'exploitation |
| 210 | Ouvre l'éditeur de type DFB |
| 212 | Ouvre le navigateur application |
| 214 | Ouvre le navigateur de mise au point Grafcet |
| 215 | Ouvre l'éditeur de code DFB en ST |

SetPosPL7Windows

Généralités Cette fonction permet de dimensionner la fenêtre PL7.

Description **Syntaxe :**

integer **PosPL7Windows**(integer CoordX, integer CoordY, integer CoordCX, integer CoordCY)

- **Entrée :**
Les coordonnées x, y de la fenêtre.
- **Fonction appelée :**
MoveWindow: Fonction mfc.
- **Contexte nominal :**
Le serveur PL7 avec IHM est actif.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_ACTION | Echec de la commande. |
| SRV_ERR_GEN_PARAM_EMPTY | Paramètre vide. |
| SRV_ERR_GEN_WITHIHM | Fonction disponible avec le serveur PL7 en mode IHM. |

- **Sortie :**
Code retour de type `short`.

ShowProgram

Généralités

Cette fonction permet d'ouvrir les programmes présents dans la chaîne de paramètres passée en paramètre.

Description

Syntaxe :

```
integer ShowProgram(String p_lpListProgram)
```

- **Entrée :**
Saisir les chaînes de caractères contenant la liste des programmes que le serveur doit ouvrir.
exemple de format : mast\lad1;mast\lit2;evt\evt0;sr1;mast\prl
- **Fonction appelée :**
`ShowProgram` : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 avec IHM est actif.
Une application est ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_ACTION | Echec de la commande. |
| SRV_ERR_GEN_NOTOPEN | Pas d'application ouverte. |
| SRV_ERR_GEN_PARAM_EMPTY | Paramètre vide. |
| SRV_ERR_GEN_WITHIHM | Fonction disponible avec le serveur PL7 en mode IHM. |
| SRV_ERR_GEN_SHOWPROGRAM | Impossible d'ouvrir le module. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
 - **Limitations :**
Cette fonction n'est pas capable d'ouvrir un DFB non instancié. Pour visualiser le code d'un DFB non instancié il faut utiliser la fonction `ShowDFB`.
-

CloseProgram

Généralités Cette fonction permet de fermer les programmes présents dans la chaîne de caractères passée en paramètre.

Description

Syntaxe :

```
integer CloseProgram(String p_lpListProgram)
```

- **Entrée :**
Chaînes de caractères contenant la liste des programmes que le serveur doit fermer.
Exemple de format : mast\lad1;mast\lit2;evt\evt0;sr1;mast\prl.
- **Fonction appelée :**
CloseProgram: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 avec IHM est actif.
Une application est ouverte.
- **Cas d'erreur :**

| | |
|---------------------------|--|
| SRV_ERR_GEN_NOTOPEN | Pas d'application ouverte. |
| SRV_ERR_GEN_ACTION | Echec de fermeture. |
| SRV_ERR_GEN_PARAM_EMPTY | Paramètre vide. |
| SRV_ERR_GEN_WITHIHM | Fonction disponible avec le serveur PL7 en mode IHM. |
| SRV_ERR_GEN_CLOSEPROGRAMM | Impossible de fermer le module. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
- **Limitations :**
Cette fonction n'est pas capable de fermer un DFB ouvert.

ShowIOModule

Généralités

Cette fonction permet d'ouvrir les modules d'entrée/sortie présents dans la chaîne de caractères passée en paramètre.

Description

- **Entrée :**
Chaînes de caractères contenant la liste des modules que le serveur doit ouvrir. Le format de la chaîne de caractères est `rack,module; rack,module`
exemple de format : `0,0;0,1;1,2`
- **Fonction appelée :**
`ShowIOModule`: Fonction de la gestion station (`gesta.dll`).
- **Contexte nominal :**
Le serveur PL7 avec IHM est actif.
Une application est ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_NOTOPEN | Pas d'application ouverte. |
| SRV_ACTION_NOK | Problème d'activation des droits utilisateurs. |
| SRV_ERR_GEN_PARAM_EMPTY | Paramètre vide. |
| SRV_ERR_GEN_WITHIHM | Fonction disponible avec le serveur PL7 en mode IHM. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
 - **Limitations :**
L'ouverture d'un module IO se fait en mode local, l'écran de configuration du module est ouvert. Si l'utilisateur passe du mode local en mode connecté, le serveur ne bascule pas de l'écran configuration à l'écran mise au point.
L'utilisateur peut soit agir directement sur le pl7pro soit faire une fermeture/ouverture du module à l'aide du client.
-

ClosetIOModule

Généralités Cette fonction permet de fermer les modules d'entrée/sortie présents dans la chaîne de caractères passée en paramètre.

Description

Syntaxe

integer **ClosetIOModule**(String p_lpListIOModule).

- **Entrée :**
Chaînes de caractères contenant la liste des modules que le serveur doit ouvrir.
exemple de format : 0,0;0,1;1,2
- **Fonction appelée :**
CloseIOModule: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 avec IHM est actif.
Une application est ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_ACTION | Echec de la commande de fermeture. |
| SRV_ERR_GEN_PARAM_EMPTY | Paramètre vide. |
| SRV_ERR_GEN_NOTOPEN | Pas d'application ouverte. |
| SRV_ERR_GEN_WITHIHM | Fonction disponible avec le serveur PL7 en mode IHM. |

- **Sortie :**
Code retour de type short. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
- **Limitations :**
Cette fonction ne ferme pas l'éditeur de configuration ouvert à l'aide de la fonction ShowIOModule.

ShowDFB

Généralités

Cette fonction permet de visualiser le code d'un ou plusieurs DFB présents dans la chaîne de caractères passée en paramètre. L'ouverture d'un DFB via cette fonction ne nécessite pas d'instanciation.

Description

Syntaxe

| |
|--|
| integer ShowDFB (String p_lpListeDFB) |
|--|

- **Entrée :**
Chaînes de caractères contenant la liste des DFB que le serveur doit ouvrir.
- **Fonction appelée :**
`ShowDFB`: Fonction de la gestion station (`gesta.dll`).
- **Contexte nominal :**
Le serveur PL7 avec IHM est actif.
Une application est ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_PARAM_EMPTY | Paramètre vide |
| SRV_ERR_GEN_NOTOPEN | Pas d'application ouverte |
| SRV_ERR_GEN_ACTION | Problème lors de l'ouverture d'un DFB. |
| SRV_ERR_GEN_WITHIHM | Fonction disponible avec le serveur PL7 en mode IHM. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.
-

CloseDFB

Généralités

Cette fonction permet de fermer un ou plusieurs DFB présents dans la chaîne de caractères passée en paramètre.

Description

Syntaxe

```
integer CloseDFB(String p_lpListeDFB)
```

- **Entrée :**
Chaînes de caractères contenant la liste des DFB que le serveur doit fermer.
- **Fonction appelée :**
CloseDFB: Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
Le serveur PL7 avec IHM est actif.
Une application est ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_PARAM_EMPTY | Paramètre vide |
| SRV_ERR_GEN_NOTOPEN | Pas d'application ouverte |
| SRV_ERR_GEN_ACTION | Problème lors de l'ouverture d'un DFB. |
| SRV_ERR_GEN_WITHIHM | Fonction disponible avec le serveur PL7 en mode IHM. |

- **Sortie :**
Code retour de type `short`. Soit 0 dans le cas d'un chargement réussi, soit un code erreur.

GetMessageError

Généralités

Cette fonction permet d'associer un message d'erreur en fonction du code erreur passé en paramètre.

Description

Syntaxe

```
String GetMessageError(integer p_iCodeError)
```

- **Entrée :**
Le code erreur d'un des services à retourner.
 - **Contexte nominal :**
Le serveur PL7 avec est actif.
 - **Sortie :**
Une chaîne de caractères correspondant au libellé de l'erreur.
-

GetServerVersion

Généralités Cette fonction permet de connaître le numéro de version du serveur PL7pro.

Description **Syntaxe**

| |
|----------------------------------|
| String GetServerVersion() |
|----------------------------------|

- **Contexte nominal :**
Le serveur PL7 avec est actif.
 - **Sortie :**
Une chaîne de caractères correspondant au libellé du numéro de version du serveur PL7pro.
-

ImportScyFile

Généralités

Cette fonction permet d'importer à partir d'un fichier scy des symboles dans l'application active.

Description

Syntaxe :

| |
|---|
| integer ImportScyFile (string ScyFile) |
|---|

- **Entrée :**
Saisir 1 chaînes de caractères contenant le nom et le chemin du fichier scy que l'on va importer.
- **Fonction appelée :**
ImportScyFile : Fonction de la gestion station (gesta.dll).
- **Contexte nominal :**
 - Le serveur PL7 est actif,
 - PL7 est en mode local, station ouverte.
- **Cas d'erreur :**

| | |
|-------------------------|--|
| SRV_ERR_GEN_ACTION | Erreur lors de l'exécution de la fonction. |
| SRV_ERR_GEN_PARAM_EMPTY | Le nom du fichier dat est incorrect ou vide. |
| SRV_ERR_GEN_NOTOPEN | Pas d'application STX ouverte. |
| SRV_ERR_SRC_FILE | Problème sur le fichier source. |

- **Sortie :**
Code retour de type short. Soit 0 dans le cas d'une importation réussie.
 - **Limitations :**
Le serveur de PL7, sans IHM, une fois l'importation réussie, ne peut pas fonctionner en mode avec IHM sans une sauvegarde préalable de l'application ouverte.
-

Performances

6

Présentation

Objet de ce chapitre

Ce chapitre décrit les performances des instructions du langage PL7. Il permet ainsi de calculer le temps d'exécution d'une application et la taille mémoire occupée.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

| Sous-chapitre | Sujet | Page |
|---------------|---|------|
| 6.1 | Informations générales | 126 |
| 6.2 | Performances des instructions sur automates Micro | 128 |
| 6.3 | performances des instructions sur automates Premium | 159 |
| 6.4 | Fonctions avancées | 198 |

6.1 Informations générales

Principes de calcul

Durée d'exécution du programme application

Le calcul du temps d'exécution du programme s'effectue à l'aide des tableaux pages suivantes, en faisant la somme des temps pour chaque instruction du programme.

Note : le temps obtenu correspond à un temps maximal. En effet un bloc opération, un sous programme ne seront traités que si la condition d'exécution (équation logique conditionnant l'exécution du bloc, du sous programme) est vrai, il se peut donc que le temps effectif soit très inférieur au temps maximal calculé.

Le calcul du temps de cycle complet fait intervenir des paramètres spécifiques à l'automate (temps d'overhead, durée d'échange d'entrées/sorties...). Pour la procédure de calcul complète se référer au manuel de mise en oeuvre de l'automate concerné (chapitre performance).

Taille mémoire application

La taille de l'application est la somme des éléments suivants :

| Élément | Méthode de calcul |
|-------------------------------------|---|
| Programme | Effectuez la somme de chacune des instructions du programme et multiplier par le coefficient correspondant au langage utilisé (voir page suivante) |
| Fonctions avancées | Voir <i>Taille mémoire des fonctions avancées, p. 210</i> |
| Objets PL7 configurés | Voir <i>Taille mémoire des objets, p. 200</i> |
| Module d'entrées/sorties configurés | Voir <i>Bilan des consommations mémoire des modules sur Micro, p. 201</i> et <i>Bilan des consommations mémoire des modules sur Premium, p. 204</i> |
| Commentaires | Les commentaires programme occupent 1 octet par caractère. |

Dans les tableaux des pages suivantes, les informations sur les tailles concernent le volume de codes instructions.

Pour connaître la taille totale d'une instruction ou d'un programme, il faut appliquer un coefficient multiplicateur qui tient compte des informations typiques par langage.

| Langage | Taille |
|----------------------|---|
| Langage à contacts | Volume total = 1,7 x Volume de code |
| Littéral structuré | Volume total = 1,6 x Volume de code |
| Liste d'instructions | Pour automate Micro : Volume total = 1,4 x Volume de code |
| | Pour automate Premium : Volume total = 1,6 x Volume de code |
| Grafcet | Volume Graphe (en mots) = $214 + 17 * \text{nb étapes graphe} + 2 * \text{nb total étapes configurées} + 4 * \text{nb actions programmées}$ |

Note : Les chiffres indiqués dans les tableaux suivants sont des estimations moyennes obtenues à partir d'une application type. Il n'est pas possible de fournir d'informations exactes, car PL7 optimise l'utilisation mémoire, en fonction du contenu et de la structure de l'application.

Le chapitre *Description des zones mémoire*, p. 199 rappelle les différentes zones mémoires occupées par l'application.

6.2 Performances des instructions sur automates Micro

Présentation

Objet de ce sous chapitre

Ce sous chapitre décrit les performances des instructions exécutées sur automate Micro.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

| Sujet | Page |
|---|------|
| Performances des instructions booléennes | 129 |
| Performances des blocs fonction | 131 |
| Arithmétique entière et à virgule flottante | 134 |
| Instructions sur programme et structures de contrôle | 138 |
| Conversions numériques | 140 |
| Instructions sur chaîne de bits | 141 |
| Instruction sur tableaux de mots, doubles mots et flottants | 144 |
| Instructions de gestion du temps | 151 |
| Instructions sur chaînes de caractères | 153 |
| Fonctions métier et fonctions Orphée | 155 |
| Instructions d'entrées/sorties explicite | 158 |

Performances des instructions booléennes

Performances Le tableau ci-après décrit les performances des instructions booléennes.

| LD | IL | ST | Objets | Temps d'exécution (µs) | | | Taille En mots |
|----|---|----|---------------|------------------------|--------------|----------------|----------------------|
| | | | | 37 05/ 08/10 | 37 20 ram | 37 20 cart. | |
| | | | | 0,25 | 0,13 | 0,19 | 1 |
| | LD LDN | | %M1 (1) | 0,25 | 0,13 | 0,19 | 1 |
| | | | %M1[%MW2] | 13,10 | 12,85 | 12,85 | 7 |
| | | | %MW0:X0 (2) | 6,06 | 5,75 | 5,75 | 4 |
| | | | %lWi.j:Xk (3) | 77,04 | 69,25 | 69,25 | 8 |
| | | | %MW0[%MW8]:X0 | 16,29 | 15,55 | 15,55 | 8 |
| | | | %KW0[%MW8]:X0 | 87,27 | 79,05 | 79,05 | 12 |
| | LDR, LDF | | %M1 | 0,50 | 0,25 | 0,38 | 2 |
| | | | %M1[%MW2] | 13,01 | 12,75 | 12,75 | 7 |
| | AND, ANDN , AND (, AND (N , idem OR | | | idem LD,LDN | | | |
| | | | | idem LD,LDN | | | |
| | ANDR, ANDF, AND (R, AND (F, idem OR | | | idem LDR,LDF | | | |
| | | | | idem LDR,LDF | | | |
| | XOR XORN | | %M1 | 1,25 | 0,63 | 0,94 | 5 |
| | | | %M1[%MW2] | 26,94 | 26,08 | 26,26 | 13 |
| | | | %MW0:X0 | 12,86 | 11,88 | 12,06 | 10 |
| | | | %lWi.j:Xk | 83,84 | 75,38 | 75,56 | 14 |
| | | | %MW0[%MW8]:X0 | 33,33 | 31,48 | 31,66 | 14 |
| | | | %KW0[%MW8]:X0 | 104,31 | 94,98 | 95,16 | 18 |
| | XORR, XORF | | %M1 | 2,25 | 1,13 | 1,69 | 9 |
| | | | %M1[%MW2] | 27,28 | 26,13 | 26,44 | 19 |

| LD | IL | ST | Objets | Temps d'exécution (µs) | | | Taille En mots |
|--|----------------------------|-----|---------------------------------|------------------------|--------------|----------------|----------------------|
| | | | | 37 05/ 08/10 | 37 20 ram | 37 20 cart. | |
| —()— —(/)— —(S)— —(R)— | ST STN S R | | %M1 (1) | 0,50 | 0,25 | 0,38 | 2 |
| | | | %M1[%MW2] | 13,10 | 12,85 | 12,85 | 7 |
| | | | %MW0:X0 | 5,88 | 5,60 | 5,60 | 4 |
| | | | %NW{i}.j.Xk (3) | 76,86 | 69,10 | 69,10 | 8 |
| | | | %MW0[%MW8]:X0 | 16,41 | 15,65 | 15,65 | 8 |
| bobines multiples en ladder, « coût » de la 2ème bobine et des suivantes | | | | 0,25 | 0,13 | 0,19 | 1 |
| bloc opération | [action] | | bloc exécuté | 0,74 | 0,75 | 0,75 | 1 |
| | | | non exécuté | 5,55 | 5,40 | 5,40 | 1 |
| bloc comparaison horizontal | [LD [comparaison]] | | Temps en plus de la comparaison | 0,00 | 0,00 | 0,00 | 0 |
| bloc opération vertical | | | entre 2 %MWi | 12,38 | 11,85 | 11,85 | 4 |
| Convergence | [] | [] | bloc exécuté | 0,25 | 0,13 | 0,19 | 1 |
| Divergence non suivie d'une convergence | [] | | Ladder, 1 divergence | 0,25 | 0,13 | 0,19 | 1 |
| | [MPS, MPP, MRD] | | List MPS+MPP | 0,75 | 0,38 | 0,56 | 3 |
| | | | List MRD | 0,25 | 0,13 | 0,19 | 1 |

Légende

1. Ceci concerne tous les objets bits forçables : %I, %Q, %X, %M, %S,
2. autres objets de même type : bits de sortie de bloc fonction %TMi.Q ..., bits extraits de mots système %SWi:Xj
3. autres objets de même type : bits extraits de mots communs %NW{i}.j.Xk, bits extraits de mots d'E/S %IWi.j.Xk, %QWi.j.Xk, bits extraits de %KW, bits de défaut %li.j.ERR.

Performances des blocs fonction

Temporisateur IEC

Le tableau ci-après décrit les performances du bloc fonction temporisateur IEC

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | Taille (mots) |
|-------------------------|----------------------------|------------|-----------------|------------------------|-------|---------------|
| | | | | 3705/08/10 | 3720 | |
| front montant sur IN | IN %TM1 (front montant) | START %TM1 | lancement timer | 43,39 | 41,11 | 3 |
| front descendant sur IN | IN %TM1 (front descendant) | DOWN %TM1 | arrêt timer | 17,47 | 17,01 | |
| IN =1 | IN %TM1 (=1) | | timer actif | 18,74 | 17,99 | |
| IN =0 | IN %TM1 (=0) | | timer inactif | 17,40 | 16,67 | |

Temporisateur PL7-3

Le tableau ci-après décrit les performances du bloc fonction temporisateur PL7-3

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | Taille (mots) |
|------|----|-----------|---------------|------------------------|-------|---------------|
| | | | | 3705/08/10 | 3720 | |
| | | START %T1 | armer | | | 3 |
| | | STOP %T1 | geler | 12,63 | 12,15 | |
| E =0 | | RESET %T1 | reset | 12,94 | 12,15 | |
| | | | timer actif | 17,55 | 17,00 | |
| | | | timer inactif | | | |

**Compteur/
décompteur**

Le tableau ci-après décrit les performances du bloc fonction compteur/décompteur

| LD | IL | ST | Condition s | Temps d'exécution (µs) | | Taille (mots) |
|----------------------------|--------------------------|---------------|------------------|------------------------|-------|------------------|
| | | | | 3705/08/10 | 3720 | |
| reset, R=1 | R %C8 (=1) | RESET %C8 | reset | 18,69 | 17,92 | 3 |
| preset, S=1 | S %C9 (=1) | PRESET %C9 | preset | 20,42 | 19,73 | |
| front montant sur CU | CU %C8 (front) | UP %C8 | up | 19,92 | 19,10 | |
| front montant sur CD | CD %C9 (front) | DOWN %C9 | down | 19,92 | 19,10 | |
| entrées inactives | R/S/CU/CD bit inactif | | aucune action | 13,27 | 12,81 | |

Monostable

Le tableau ci-après décrit les performances du bloc fonction monostable

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | Taille (mots) |
|---------------------------|-----------------------------|---------------|---------------------|------------------------|-------|------------------|
| | | | | 3705/08/10 | 3720 | |
| front montant sur S | S %MN0, front montant | START %MN0 | lancement | 35,08 | 33,16 | 3 |
| S=1 | S %MN0, S =1/0 | | monostable actif | 11,64 | 11,17 | |

Registre

Le tableau ci-après décrit les performances du bloc fonction registre

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | Taille (mots) |
|----------------------|-----------------------|--------------|--------------|---------------------------|-------|------------------|
| | | | | 3705/08/10 | 3720 | |
| front sur I | I %R2 (front) | PUT %R2 | stockage | 21,90 | 21,27 | 3 |
| front sur O | O %R2 (front) | GET %R2 | déstockage | 21,90 | 21,27 | |
| R=1 | R %R1 (=1) | RESET %R2 | reset | 16,90 | 16,02 | |
| entrées inactives | I/O/R, bit inactif | | pas d'action | 12,61 | 12,19 | |

**Programmeur
cyclique**

Le tableau ci-après décrit les performances du programmeur cyclique

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | Taille (mots) |
|-------------------|------------------|------------|--------------------|------------------------|--------|---------------|
| | | | | | | |
| front sur U | U %DR0 | UP %DR1 | up, fixe | 181,37 | 169,13 | 3 |
| | | | par bit d'ordre | 19,30 | 19,30 | |
| R=1 | R %DR1 | RESET %DR2 | reset, fixe | 174,15 | 162,03 | |
| | | | par bit d'ordre | 19,30 | 19,30 | |
| entrées inactives | R/U, bit inactif | | pas d'action, fixe | 175,92 | 164,00 | |
| | | | par bit d'ordre | 19,30 | 19,30 | |

Arithmétique entière et à virgule flottante

Corrections en fonction du type d'objets

Les temps et volumes pages ci-après sont donnés pour des objets de type %MW0, %MD0 ou %MF0.

Le tableau suivant décrit les corrections à apporter aux valeurs données dans le tableau de performances des instructions arithmétiques si d'autres types d'objets sont utilisés.

| Type d'objet | Position de l'objet | Type de correction | Temps d'exécution (µs) | | Taille en mots |
|---|--|---|------------------------|-------|----------------|
| | | | 3705/08/10 | 3720 | |
| Valeur immédiate simple longueur | - | Valeur à enlever à celle mentionnée pour %MW | 1,20 | 1,10 | 0 |
| Valeur immédiate double longueur | - | Valeur à enlever à celle mentionnée pour %MD ou %MF | 0,75 | 1 | 0 |
| Mots, doubles Mots ou flottants indexés | Après le signe := | Valeur à ajouter | 10,52 | 10,05 | 4 |
| | Première opération, le premier opérande n'étant pas indexé, ou affectation | Valeur à ajouter | 11,20 | 10,60 | 5 |
| | Deuxième opérande si le premier opérande est également indexé | Valeur à ajouter | 13,37 | 12,60 | 5 |
| %KWi, %KWi[%MWj] %KDi, %KFi mots commun, mots d'entrées/ sorties | - | Valeur à ajouter | 70,98 | 63,50 | 2 |

Correction en fonction du contexte de l'opération

Le tableau suivant décrit les corrections à apporter aux valeurs données dans le tableau de performances des instructions arithmétiques en fonction du contexte de l'opération.

| Contexte de l'opération | Type d'objet | Type de correction | Temps d'exécution (µs) | | Taille 37xx |
|--|--------------|---|------------------------|------|----------------|
| | | | 3705/08/10 | 3720 | |
| L'opération est au minimum en deuxième position dans la phrase Ex : %MW2 dans :=%MW0*%MW1*%MW2 | %MW | Valeur à ajouter à celle mentionnée pour %MW | 0,69 | 0,55 | 0 |
| | %MD ou %MF | Valeur à ajouter à celle mentionnée pour %MD ou %MF | 0,99 | 0,75 | 0 |
| Opération avec résultat d'une opération entre parenthèse ou plus prioritaire Ex : %MW0+%MW2+(...) | %MW | Valeur à ajouter à celle mentionnée pour %MW | 2,86 | 2,55 | 1 |
| | %MD ou %MF | Valeur à ajouter à celle mentionnée pour %MD ou %MF | 3,60 | 3,15 | 1 |

Tableau de performance des instructions

Le tableau ci-après décrit les performances des instructions arithmétiques.

| ST | Objets | Conditions | Temps d'exécution (µs) | | Taille (mots) 37xx |
|---------------------|-----------------|------------|------------------------|--------|-----------------------|
| | | | 3705/08/10 | 3720 | |
| objet après le := | %MW0 | | 4,81 | 4,50 | 2 |
| | %MD0,%MF0 | | 6,45 | 5,70 | 2 |
| := | %MW0 | | 4,46 | 4,30 | 2 |
| | %MD0 et %MF0 | | 5,15 | 4,85 | 2 |
| =, <>, <=, <, >, >= | %MW0 | | 8,94 | 8,50 | 4 |
| | %MD0 | | 10,71 | 10,26 | 4 |
| | %MF0 | | 29,06 | 28,39 | 4 |
| AND, OR, XOR | %MW0 | | 7,29 | 6,90 | 3 |
| | %MD0 | | 9,21 | 8,55 | 3 |
| +, - | %MW0 | | 7,29 | 6,90 | 3 |
| | %MD0 | | 9,21 | 8,55 | 3 |
| | %MF0 | | 62,83 | 61,20 | 3 |
| * | %MW0 | | 9,75 | 9,10 | 3 |
| | %MD0 | | 39,63 | 36,50 | 3 |
| | %MF0 | | 58,26 | 56,90 | 3 |
| /, REM | %MW0 | | 10,69 | 10,08 | 3 |
| | %MD0 | | 205,21 | 201,38 | 3 |
| / | %MF0 | | 62,47 | 60,25 | 3 |
| ABS, -objet | %MW0 | | 7,20 | 6,95 | 3 |
| | %MD0 | | 9,97 | 9,53 | 3 |
| | %MF0 | | 13,01 | 12,50 | 3 |
| NOT | %MW0 | | 6,69 | 6,45 | 3 |
| | %MD0 | | 7,80 | 7,40 | 3 |
| SQRT | %MW0 | | 17,02 | 16,70 | 3 |
| | %MD0 | | 85,73 | 85,25 | 3 |
| | %MF0 | | 165,04 | 158,40 | 3 |
| INC, DEC | %MW0 | | 4,86 | 4,40 | 2 |
| | %MD0 | | 5,20 | 4,75 | 2 |

| ST | Objets | Conditions | Temps d'exécution (μs) | | Taille (mots) 37xx |
|--------------------|--------|------------------------|------------------------|---------|--------------------|
| | | | 3705/08/10 | 3720 | |
| SHL, SHR, ROL, ROR | %MW0 | pour 1 bit | 17,74 | 17,05 | 5 |
| | %MD0 | pour 1 bit | 20,58 | 19,15 | 5 |
| | | par bit supplémentaire | 0,063 | | |
| LN | %MF0 | | 1371,60 | 1270,00 | 3 |
| LOG | %MF0 | | 1458,00 | 1350,00 | 3 |
| EXP | %MF0 | | 1155,60 | 1070,00 | 3 |
| EXPT | %MF0 | | 2988,00 | 2490,00 | 3 |
| TRUNC | %MF0 | | 204,00 | 170,00 | 3 |
| COS | %MF0 | | 2829,60 | 2620,00 | 3 |
| SIN | %MF0 | | 2840,40 | 2630,00 | 3 |
| TAN | %MF0 | | 2937,60 | 2720,00 | 3 |
| ACOS | %MF0 | | 4082,40 | 3780,00 | 3 |
| ASIN | %MF0 | | 4082,40 | 3780,00 | 3 |
| ATAN | %MF0 | | 2786,40 | 2580,00 | 3 |
| DEG_TO_RAD | %MF0 | | 852,00 | 710,00 | 3 |
| RAD_TO_DEG | %MF0 | | 720,00 | 600,00 | 3 |

Instructions sur programme et structures de contrôle

Performance des instructions sur programme

Le tableau ci-après décrit les performances des instructions sur programme.

| ST | Temps d'exécution (μ s) | | Taille (mots) 37xx |
|-----------|------------------------------|-------|-----------------------|
| | 3705/08/10 | 3720 | |
| Jump %Li | 41,93 | 38,20 | 3 |
| Maskevt | 12,21 | 10,80 | 1 |
| Unmaskevt | 40,27 | 37,10 | 1 |
| SRI | 48,68 | 42,88 | 3 |
| Return | 42,18 | 38,33 | 3 |

**Performance des
Structures de
contrôle**

Le tableau ci-après décrit les performances des instructions de type structures de contrôle.

| ST | | Temps d'exécution (µs) | | Taille (mots) 37xx |
|--|---|------------------------|-------|-----------------------|
| | | 3705/08/10 | 3720 | |
| <cond> | évaluation de la condition | | | |
| bit forçable | voir instruction booléenne LD %M1 | | | |
| comparaison | voir comparaisons =,<,> ... | | | |
| if <cond > then <action> end_if; | les temps et volumes indiqués ci-dessous sont à ajouter à ceux de l'action contenue dans la structure | | | |
| condition vraie | | 3,60 | 3,30 | 2 |
| condition fausse (saut) | | 5,55 | 5,40 | |
| If <cond> then <action1> else <action2> end_if; | | | | |
| condition vraie | | 9,15 | 8,70 | 4 |
| condition fausse | | 5,55 | 5,40 | |
| while <cond> do.<action> end_while | | | | |
| passage dans la boucle avec rebouclage | | 9,15 | 8,70 | 2 |
| sortie de la boucle | | 5,55 | 5,40 | |
| repeat <action> until <cond> end_repeat | | | | |
| passage dans la boucle avec rebouclage | | 5,55 | 5,40 | 2 |
| dernier passage | | 3,60 | 3,30 | |
| for <mot1:=mot2>to <mot3> do <action> end_for | | | | |
| entrée dans le for, exécuté une seule fois | | 8,58 | 8,25 | 15 |
| passage dans la boucle avec rebouclage | | 29,38 | 27,35 | |
| sortie de la boucle | | 20,42 | 19,40 | |

Conversions numériques

Performances

Le tableau ci-après décrit les performances des instructions de conversions numériques.

| ST | Temps d'exécution (µs) | | | Volume (mots) |
|--------------|------------------------|----------|-----------|---------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| BCD_TO_INT | 25,03 | 24,55 | 24,55 | 3 |
| INT_TO_BCD | 21,66 | 21,15 | 21,15 | 3 |
| GRAY_TO_INT | 36,98 | 36,55 | 36,55 | 3 |
| INT_TO_REAL | 40,90 | 40,75 | 40,75 | 3 |
| DINT_TO_REAL | 33,32 | 32,55 | 32,55 | 3 |
| REAL_TO_INT | 58,75 | 58,55 | 58,55 | 3 |
| REAL_TO_DINT | 44,59 | 44,05 | 44,05 | 3 |
| DBCD_TO_DINT | 1 324,85 | 1 065,15 | 1 134,70 | 5 |
| DBCD_TO_INT | 1 265,54 | 925,70 | 986,15 | 5 |
| DINT_TO_DBCD | 1 124,85 | 825,15 | 879,10 | 5 |
| INT_TO_DBCD | 564,85 | 445,15 | 474,40 | 5 |

Instructions sur chaîne de bits

Initialisation d'un tableau de bits

Le tableau ci-après décrit les performances des instructions d'initialisation d'un tableau de bits.

| ST | Taille (bit) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|--------------|--------------|------------------------|----------|-----------|-----------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %M30:8 := 0 | 8 | 19,38 | 18,88 | 18,88 | 6 |
| %M30:16 := 1 | 16 | 20,38 | 19,88 | 19,88 | 6 |
| %M30:24 := 2 | 24 | 24,25 | 23,35 | 23,35 | 6 |
| %M30:32 := 2 | 32 | 25,25 | 24,35 | 24,35 | 6 |

Copie d'un tableau de bits dans un tableau de bits

Le tableau ci-après décrit les performances des instructions de copie d'un tableau de bits dans un autre tableau de bits.

| ST | Taille (bit) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|---------------------------------|--------------|------------------------|----------|-----------|-----------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %M30:8 := %M20:8 | 8 | 25,54 | 24,79 | 24,79 | 6 |
| %M30:16 := %M20:16 | 16 | 26,16 | 25,41 | 25,41 | 6 |
| %M30:24 := %M20:24 | 24 | 33,41 | 32,26 | 32,26 | 6 |
| %M30:32 := %M20:32 | 32 | 35,91 | 34,76 | 34,76 | 6 |
| %M30:16 := COPY_BIT(%M20:16) | 16 | 281,63 | 230,00 | 244,95 | 9 |
| | 32 | 440,82 | 360,00 | 383,40 | 9 |
| | 128 | 1261,22 | 1030,00 | 1096,95 | 9 |

Instructions logiques sur tableau de bits

Le tableau ci-après décrit les performances des instructions logiques sur tableau de bits.

| ST | Taille (bits) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|---------------------------------------|---------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| AND_ARX, OR_ARX, XOR_ARX | | | | | |
| %M0:16 := AND_ARX(%M30:16,%M50:16) | 16 | 397,42 | 320,00 | 340,80 | 12 |
| %M0:32 := AND_ARX(%M30:32,%M50:32) | 32 | 620,97 | 500,00 | 532,50 | 12 |
| %M0:128 := AND_ARX(%M30:128,%M50:128) | 128 | 1 887,74 | 1 520,00 | 1 618,80 | 12 |
| NOT_ARX | | | | | |
| %M0:16 := NOT_ARX(%M30:16) | 16 | 281,63 | 230,00 | 244,95 | 9 |
| | 32 | 440,82 | 360,00 | 383,40 | 9 |
| | 128 | 1261,22 | 1030,00 | 1096,95 | 9 |

Copie d'un tableau de bits dans un tableau de mots

Le tableau ci-après décrit les performances des instructions de copie de tableaux de bits dans un tableau de mots.

| ST | Taille (bits) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|---------------------------------|---------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW1 := %M30:8 | 8 | 14,84 | 14,36 | 14,36 | 5 |
| %MW1 := %M30:16 | 16 | 16,34 | 15,86 | 15,86 | 5 |
| %MD2 := %M30:24 | 24 | 14,54 | 14,23 | 14,23 | 5 |
| %MD2 := %M30:32 | 32 | 16,04 | 15,73 | 15,73 | 5 |
| %MW1:4 := BIT_W(%M40:80,0,17,2) | 17 | 501,43 | 390,00 | 415,35 | 16 |
| %MD1:4 := BIT_D(%M30:80,0,33,0) | 33 | 379,53 | 530,00 | 564,45 | 16 |

**Copie d'un
tableau de mots
dans un tableau
de bits**

Le tableau ci-après décrit les performances des instructions de copie de tableaux de mots dans un tableau de bits.

| ST | Taille (bits) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|-------------------------------------|------------------|------------------------|----------|-----------|--------------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %M30:8 := %MW1 | 8 | 19,28 | 18,68 | 18,68 | 5 |
| %M30:16 := %MW2 | 16 | 20,28 | 19,68 | 19,68 | 5 |
| %M30:24 := %MD1 | 24 | 21,20 | 20,37 | 20,37 | 5 |
| %M30:32 := %MD3 | 32 | 22,20 | 21,37 | 21,37 | 5 |
| %M30:32 := W_BIT(%MW200:2,0,2,0) | 32 | 488,68 | 370,00 | 394,05 | 16 |
| %M30:32 := D_BIT(%MD0:1,0,2,0) | 32 | 567,33 | 460,00 | 489,90 | 16 |

Instruction sur tableaux de mots, doubles mots et flottants

Initialisation d'un tableau de mots par un mot

Le tableau ci-après décrit les performances des instructions d'initialisation d'un tableau de mots par un mot..

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|-------------------|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW0:10 := %MW100 | 10 mots | 47,46 | 42,15 | 42,15 | 7 |
| | par mot | 0,34 | 0,20 | 0,20 | |
| %MD0:10 := %MD100 | 10 double mots | 81,27 | 74,45 | 74,45 | 7 |
| | par double mot | | 2,87 | 2,65 | 2,65 |

Copie d'un tableau de mots dans un tableau de mots

Le tableau ci-après décrit les performances des instructions de copie d'un tableau de mots dans un autre tableau de mots.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|--------------------|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW0:10:=%MW20:10; | 10 mots | 95,80 | 85,35 | 85,35 | 9 |
| | par mot | 0,77 | 0,50 | 0,50 | |
| %MD0:10:=%MD20:10; | 10 double mots | 111,13 | 97,65 | 97,65 | 9 |
| | par double mot | 1,54 | 1,00 | 1,00 | |

**Instructions
arithmétiques et
logiques entre 2
tableaux de mots**

Le tableau ci-après décrit les performances des instructions arithmétiques et logiques entre 2 tableaux de mots.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|------------------------------------|----------------------------------|------------------------|----------|-----------|--------------------------|
| | | 3705/08/ 10 | 3720 ram | 3720 cart | |
| +, - | | | | | |
| %MW0:10 := %MW10:10 + %MW20:10; | 10 mots | 168,04 | 151,95 | 151,95 | 14 |
| | par mot | 7,13 | 6,35 | 6,35 | |
| %MD0:10:= %MD10:10+%MD20:10; | 10 double mots | 239,17 | 214,40 | 214,40 | 14 |
| | par double mot | 13,84 | 12,25 | 12,25 | |
| * | | | | | |
| %MW0:10:= %MW10:10 * %MW20:10; | 10 mots | 189,32 | 175,40 | 175,40 | 14 |
| | par mot | 9,27 | 8,70 | 8,70 | |
| %MD0:10:= %MD10:10 * %MD20:10; | 10 double mots | 710,35 | 603,80 | 603,80 | 14 |
| | par double mot | 61,64 | 51,20 | 51,20 | |
| /, REM | | | | | |
| %MW0:10:= %MW10:10 / %MW20:10; | 10 mots | 224,76 | 181,40 | 181,40 | 14 |
| | par mot | 13,14 | 9,30 | 9,30 | |
| %MD0:10:= %MD10:10 / %MD20:10; | 10 double mots | 2 192,38 | 2 157,35 | 2 157,35 | 14 |
| | par double mot | 209,16 | 206,55 | 206,55 | |
| AND, OR, XOR | | | | | |
| %MW0:10:=%MW10:10 AND %MW20:10; | 10 mots | 163,69 | 147,40 | 147,40 | 14 |
| | par mot | 6,66 | 5,85 | 5,85 | |
| %MD0:10:=%MD10:10 AND %MD20:10; | 10 double mots | 240,14 | 215,90 | 215,90 | 14 |
| | par double mot | 13,94 | 12,40 | 12,40 | |

Instructions arithmétiques et logiques entre 1 tableau de mots et 1 mot

Le tableau ci-après décrit les performances des instructions arithmétiques et logiques entre 1 tableau de mots et 1 mot.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|---|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| +, - | | | | | |
| %MW0:10 :=%MW10:10 + %MW20; ou %MW0:10 := %MW20 + %MW10:10 | 10 mots | 119,12 | 108,55 | 108,55 | 12 |
| | par mot | 2,87 | 2,65 | 2,65 | |
| %MD0:10 :=%MD10:10 + %MD20; | 10 double mots | 159,68 | 147,45 | 147,45 | 12 |
| | par double mot | 6,57 | 6,25 | 6,25 | |
| * | | | | | |
| %MW0:10 := %MW20*%MW10:10; | 10 mots | 166,86 | 132,45 | 132,45 | 12 |
| | par mot | 7,94 | 5,05 | 5,05 | |
| %MD0:10:= %MD20*%MD10:10; | 10 double mots | 587,01 | 522,95 | 522,95 | 12 |
| | par double mot | 49,18 | 43,80 | 43,80 | |
| /, REM | | | | | |
| %MW0:10 :=%MW10:10 / %MW30; | 10 mots | 196,69 | 155,85 | 155,85 | 15 |
| | par mot | 10,86 | 7,30 | 7,30 | |
| %MD0:10:= MD10:10 / %MD30 | 10 double mots | 2 230,17 | 2 173,95 | 2 173,95 | 12 |
| | par double mot | 213,66 | 208,90 | 208,90 | |
| AND, OR, XOR | | | | | |
| %MW0:10 :=%MW10:10 AND %MW20; | 10 mots | 117,20 | 106,45 | 106,45 | 12 |
| | par mot | 2,64 | 2,40 | 2,40 | |
| %MD0:10 :=%MD20 AND %MD10:10; | 10 double mots | 587,01 | 522,95 | 522,95 | 12 |
| | par double mot | 6,47 | 6,15 | 6,15 | |
| NOT | | | | | |

| ST | Taille (du tableau de mot) | Temps d'exécution (μs) | | | Taille (mots) 37xx |
|-----------------------------|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW0:10 :=NOT(%MW10:10); | 10 mots | 110,28 | 100,25 | 100,25 | 9 |
| | par mot | 2,96 | 2,75 | 2,75 | |
| %MD0:10:=NOT(%MD10:10) | 10 double mots | 126,39 | 114,00 | 114,00 | 9 |
| | par double mot | 4,50 | 4,05 | 4,05 | |

Fonction sommation sur tableau

Le tableau ci-après décrit les performances des instructions de sommation sur tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (μs) | | | Taille (mots) 37xx |
|--------------------------|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW20:=SUM(%MW0:10); | 10 mots | 74,30 | 69,00 | 69,00 | 10 |
| | par mot | 2,44 | 2,35 | 2,35 | |
| %MD20:=SUM(%MD0:10); | 10 double mots | 83,58 | 76,90 | 76,90 | 10 |
| | par double mot | 3,17 | 2,95 | 2,95 | |
| %MF20:=SUM_ARR(%MF0:10); | 10 double mots | 1634 | 1257 | 1257 | 10 |
| | par double mot | | | | |

Fonction de comparaison de tableaux

Le tableau ci-après décrit les performances des instructions de comparaison de tableaux.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|---|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW20:= EQUAL(%MW0:10;%MW10:10); | 10 mots | 103,78 | 93,50 | 93,50 | 11 |
| | par mot | 1,13 | 0,90 | 0,90 | |
| %MD20:= EQUAL(%MD0:10;%MD10:10); | 10 double mots | 116,17 | 103,40 | 103,40 | 11 |
| | par double mot | 2,23 | 1,75 | 1,75 | |
| %MF20:= EQUAL_ARR(%MF0:10;%MF10:10); | 10 double mots | 741 | 570 | 607 | 11 |
| | par double mot | | | | |

Fonction de recherche

Le tableau ci-après décrit les performances des instructions de recherche dans un tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|--------------------------------------|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW20 := FIND_EQW(%MW0:10,%KW0) | 10 mots, cas max | 340,00 | 250,00 | 266,25 | 15 |
| %MD20 := FIND_EQD(%MD0:10, %KD0) | 10 double mots, cas max | 350,00 | 260,00 | 276,90 | 16 |
| %MF20 := FIND_EQR(%MF0:10, %KF0) | 10 double mots | 833 | 648 | 690,12 | 15 |
| %MF20 := FIND_EQRP(%MF0:10, %KF0) | 10 double mots | 845 | 650 | 692,25 | 15 |
| %MD20 := FIND_GTR(%MF0:10, %KF0) | 10 double mots | 836 | 643 | 684,79 | 15 |
| %MD20 := FIND_LTR(%MF0:10, %KF0) | 10 double mots | 836 | 643 | 684,79 | 15 |

Recherche de valeurs maxi et mini

Le tableau ci-après décrit les performances des instructions de recherche de valeurs maxi et mini dans un tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|------------------------------|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW20 := MAX_ARW(%MW0:10) | 10 mots | 350,00 | 260,00 | 276,90 | 9 |
| %MD20 := MAX_ARD(%MD0:10) | 10 double mots | 410,00 | 300,00 | 319,50 | 9 |
| %MF20 := MAX_ARR(%MF0:10) | 10 double mots | 1366 | 1051 | 1119,31 | 9 |
| %MF20 := MIN_ARR(%MF0:10) | 10 double mots | 1270 | 977 | 1040,50 | 9 |

Calcul du nombre d'occurrences

Le tableau ci-après décrit les performances des instructions du nombre d'occurrence d'une valeur dans un tableau de mots.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|---|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW20 := OCCUR_ARW(%MW0:10, %KW0) | 10 mots | 350,00 | 250,00 | 266,25 | 15 |
| %MD20 := OCCUR_ARD(%MD0:10, %KD0) | 10 double mots | 370,00 | 270,00 | 287,55 | 16 |
| %MF20 := OCCUR_ARR(%MF0:10, %KF0) | 10 double mots | 1265 | 973 | 1036,24 | 16 |

Décalage circulaire

Le tableau ci-après décrit les performances des instructions de décalage circulaire.

| ST | Taille (du tableau de mot) | Temps d'exécution (μs) | | | Taille (mots) 37xx |
|--------------------------------|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| ROL_ARW(mot ou valeur,%MWj:10) | 10 mots | 550,00 | 400,00 | 426,00 | 9 |
| ROL_ARD(%MDi,%MDj:10) | 10 double mots | 590,00 | 430,00 | 457,95 | 9 |
| ROL_ARR(%MFi,%MFj:10) | 10 double mots | 585 | 450 | 479,25 | 9 |

Instruction de tri

Le tableau ci-après décrit les performances des instructions de tri des éléments d'un tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (μs) | | | Taille (mots) 37xx |
|------------------------|----------------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| SORT_ARW(%MWi,%MWj:10) | 10 mots, cas max | 970,00 | 700,00 | 745,50 | 9 |
| SORT_ARD(%MDi,%MDj:10) | 5 double mots, cas max | 610,00 | 450,00 | 479,25 | 9 |
| SORT_ARR(%MFi,%MFj:10) | 10 double mots | 1863 | 1433 | 1526,14 | 9 |

Instructions de gestion du temps

Instructions de gestion des dates, heures et durées

Le tableau ci-après décrit les performances des instructions de gestion des dates, heures et durées.

| ST | Temps d'exécution (µs) | | | Volume (mots) |
|-------------------------------------|------------------------|----------|-----------|---------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW2:4 := ADD_DT(%MW2:4,%MD8) | 4 400,00 | 3 300,00 | 3 514,50 | 13 |
| %MD2 := ADD_TOD(%MD2,%MD8) | 2 100,00 | 1 550,00 | 1 650,75 | 9 |
| %MB2:11 := DATE_TO_STRING(%MD40) | 1 370,00 | 900,00 | 958,50 | 9 |
| %MW5 := DAY_OF_WEEK() | 220,00 | 280,00 | 298,20 | 5 |
| %MD10 := DELTA_D(%MD2, %MD4) | 1 520,00 | 1 130,00 | 1 203,45 | 9 |
| %MD10 := DELTA_DT(%MD2:4,%MW6:4) | 3 170,00 | 2 300,00 | 2 449,50 | 13 |
| %MD10 := DELTA_TOD(%MD2,%MD4) | 2 330,00 | 1 700,00 | 1 810,50 | 9 |
| %MB2:20 := DT_TO_STRING(%MW50:4) | 2 050,00 | 1 450,00 | 1 544,25 | 11 |
| %MW2:4 := SUB_DT(%MW2:4,%MD8) | 4 750,00 | 3 500,00 | 3 727,50 | 13 |
| %MD2 := SUB_TOD(%MD2,%MD8) | 2 330,00 | 1 700,00 | 1 810,50 | 9 |
| %MB2:15 := TIME_TO_STRING(%MD40) | 1 560,00 | 1 200,00 | 1 278,00 | 9 |
| %MB2:9 := TOD_TO_STRING(%MD40) | 1 270,00 | 800,00 | 852,00 | 9 |
| %MD100 := TRANS_TIME(%MD2) | 500,00 | 500,00 | 532,50 | 7 |

**Accès
horodateur**

Le tableau ci-après décrit les performances des instructions horodateur.

| ST | Temps d'exécution (μs) | | | Volume (mots) |
|---|------------------------|----------|-----------|---------------|
| | 3705/08/10 | 3720 ram | 3720 cart | 37xx |
| RRTC(%MW0:4) | 93,60 | 84,80 | 84,80 | 5 |
| WRTC(%MW0:4) | 248,61 | 230,85 | 230,85 | 5 |
| PTC(%MW0:5) | 97,98 | 88,60 | 88,60 | 5 |
| SCHEDULE(%MW0,%MW1,%MW2,%MD10,%MD12,%M0)ÿ | 1430 | 1100 | 1171,5 | 5 |

Instructions sur chaînes de caractères

Affectation, recopie de chaînes de caractères

Le tableau ci-après décrit les performances des instructions d'affectation et recopie de chaînes de caractères

| ST | Taille (caractères) | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|-------------------|---------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| %MB0:8:=%MB10:8 | 8 caractères | 105,16 | 93,80 | 93,80 | 9 |
| | par caractère | 1,65 | 1,30 | 1,30 | |
| %MB0:8:='abcdefg' | 8 caractères | 120,72 | 110,20 | 110,20 | 11 |
| | par caractère | 4,15 | 3,85 | 3,85 | 0,5 |

Conversions mot <-> chaînes de caractères

Le tableau ci-après décrit les performances des instructions de conversion de mots et de chaînes de caractères.

| ST | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|---------------------------------|------------------------|----------|-----------|--------------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| %MW1:= STRING_TO_INT(%MB0:7) | 97,69 | 91,95 | 91,95 | 7 |
| %MB0:7:= INT_TO_STRING(%MW0) | 104,36 | 96,70 | 96,70 | 7 |

Conversions double mot <-> chaînes de caractères

Le tableau ci-après décrit les performances des instructions conversion de double mots et de chaînes de caractères

| ST | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|-----------------------------------|------------------------|----------|-----------|--------------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| %MD1:= STRING_TO_DINT(%MB0:13) | 1 070,53 | 965,62 | 965,62 | 7 |
| %MB0:13:= DINT_TO_STRING(%MD0) | 322,29 | 295,35 | 295,35 | 7 |

Conversions flottant <-> chaînes de caractères

Le tableau ci-après décrit les performances des instructions conversion de flottant en de chaînes de caractères

| ST | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|-----------------------------------|------------------------|----------|-----------|--------------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| %MF1:= STRING_TO_REAL(%MB0:15) | 1 783,70 | 1 634,53 | 1 634,53 | 7 |
| %MB0:15:= REAL_TO_STRING(%MF0) | 741,75 | 681,20 | 681,20 | 7 |

Instructions de manipulation de chaînes de caractères

Le tableau ci-après décrit les performances des instructions de manipulation de chaînes de caractères

| ST | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|--|------------------------|----------|-----------|--------------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| %MB10:20 := CONCAT(%MB30:10,%MB50:10) | 1 170,00 | 770,00 | 820,05 | 15 |
| %MB10:20 := DELETE(%MB10:22,2,3); | 950,00 | 600,00 | 639,00 | 15 |
| %MW0 := EQUAL_STR(%MB10:20,%MB30:20); (le 5ème caractère diffère) | 860,00 | 520,00 | 553,80 | 13 |
| %MW0 := FIND(%MB10:20,%MB30:10); | 1 610,00 | 1 000,00 | 1 065,00 | 13 |
| %MB10:20 := INSERT(%MB30:10,%MB50:10,4); | 1 270,00 | 800,00 | 852,00 | 17 |
| %MB10:20 := LEFT(%MB30:30,20); | 920,00 | 570,00 | 607,05 | 13 |
| %MW0 := LEN(%MB10:20); | 770,00 | 340,00 | 362,10 | 9 |
| %MB10:20 := MID(%MB30:30,20,10); | 1 080,00 | 700,00 | 745,50 | 15 |
| %MB10:20 := REPLACE(%MB30:20,%MB50:10,10,10); | 1 450,00 | 870,00 | 926,55 | 19 |
| %MB10:20 := RIGHT(%MB30:30,20); | 1 480,00 | 950,00 | 1 011,75 | 13 |

Fonctions métier et fonctions Orphée

Fonctions de communication

Le tableau ci-après décrit les performances des fonctions de communication..

| ST | Temps d'exécution (µs) | | | Taille (mots) 37** |
|---|------------------------|----------|-----------|--------------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| SEND_REQ(%KW0:6,15,%MW0:1,%MW10:10,%MW30:4) | 2182 | 1818 | 1936 | 21 |
| SEND_TLG(%KW0:6,1,%MW0:5,%MW30:2) | 1636 | 1364 | 1452 | 15 |

Fonction de dialogue opérateur

Le tableau ci-après décrit les performances des instructions dialogue opérateur..

| ST | Temps d'exécution (µs) | | | Taille (mots) 37** |
|---|------------------------|----------|-----------|--------------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| SEND_MSG(ADR#1.0,%MW0:2,%MW10:2) | 2 240 | 2 000 | 2208 | 19 |
| SEND_ALARM(ADR#1.0,%MW0:2,%MW10:2) | 2 240 | 2 000 | 2208 | 19 |
| GET_MSG(ADR#1.0,%MW0:2,%MW10:2) | 2 240 | 2 000 | 2 208 | 19 |
| GET_VALUE(ADR#1.0,%MW0,%MW10:2) | 1 120 | 1 000 | 1 104 | 17 |
| ASK_MSG(ADR#1.0,%MW0:2,%MW10:2,%MW20:2) | 2 240 | 2 000 | 2 208 | 23 |
| ASK_VALUE(ADR#1.0,%MW0,%MW10:2,%MW20:2) | 2 240 | 2 000 | 2 208 | 21 |
| DISPLAY_ALRM(ADR#1.0,%MW0,%MW10:2) | 1 120 | 1 000 | 1 104 | 17 |
| DISPLAY_GRP(ADR#1.0,%MW0,%MW10:2) | 1 120 | 1 000 | 1 104 | 17 |
| DISPLAY_MSG(ADR#1.0,%MW0,%MW10:2) | 1 120 | 1 000 | 1 104 | 17 |
| CONTROL_LEDS(ADR#1.0,%MW0:2,%MW10:2) | 2 240 | 2 000 | 2 208 | 19 |
| ASSIGN_KEYS(ADR#1.0,%MW0:2,%MW10:2) | 2 240 | 2 000 | 2 208 | 19 |
| PANEL_CMD(ADR#1.0,%MW0:2,%MW10:2) | 2 240 | 2 000 | 2 208 | 19 |

Fonction de régulation

Le tableau ci-après décrit les performances des fonctions de régulation..

| ST | Condi tion | Temps d'exécution (µs) | | | Taille (mots) 37** |
|--|-----------------|------------------------|----------|-----------|--------------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| PID("PIDS1", 'Unité', %IW3.5, %MW12, %M16, %MW284:43) | deval_ mmi=0 | 1320 | 1100 | 1172 | 24 |
| | deval_ mmi=1 | 1080 | 900 | 958,5 | |
| PWM(%MW11, %Q2.1, %MW385:5) | | 600 | 500 | 532,5 | 11 |
| SERVO(%MW12, %IW3.6, %Q2.2, %Q2.3, %MW284:43, %MW390:10) | | 960 | 800 | 852 | 19 |
| PID_MMI(ADR#0.0.4, %M1, %M2:5, %MW410:62) | EN=1 | 1140 | 950 | 1012 | 20 |

Archivage de données

Le tableau ci-après décrit les performances des fonctions d'archivage de données.

| ST | Temps d'exécution (µs) | | | Taille (mots) 37** |
|-------------------------------|------------------------|----------|-----------|--------------------------|
| | 3705/08/ 10 | 3720 ram | 3720 cart | |
| SET_PCMCIA (pour 1600 mots) | | 1000 | | 24 |
| WRITE_PCMCIA (pour 1600 mots) | | 2000 | | 24 |
| READ_PCMCIA (pour 1600 mots) | | 2000 | | 24 |

Fonction Orphée Le tableau ci-après décrit les performances des fonctions de régulation..

| ST | Condition | Temps d'exécution (µs) | | | Taille (mots) 37** |
|--|---------------------|------------------------|----------|-----------|--------------------|
| | | 3705/08/10 | 3720 ram | 3720 cart | |
| DSHR_RBIT(%MD102,16,%MD204,%MD206) | écriture de 10 mots | 660 | 480 | 511 | 13 |
| DSHRZ_C(%MD102,16,%MD204,%MD206) | req miroir 10 mots | 410 | 310 | 330 | 13 |
| WSHL_RBIT(%MW102,8,%MW204,%MW206) | échange de 10 mots | 300 | 220 | 234 | 13 |
| WSHR_RBIT(%MW102,8,%MW204,%MW206) | 20 octets | 390 | 280 | 298 | 13 |
| WSHRZ_C(%MW102,8,%MW204,%MW206) | 20 octets | 300 | 220 | 234 | 13 |
| SCOUNT(%M100,%MW100,%M101,%M102,%MW101,%MW102,%M200,%M201,%MW200,%MW201) | 20 octets | 510 | 410 | 437 | 25 |

Instructions d'entrées/sorties explicite

Performances

Le tableau ci-après décrit les performances des instructions d'entrées/sorties explicites.

| ST | Temps d'exécution (µs) | | | Taille (mots) 37xx |
|---|------------------------|----------|-----------|--------------------------|
| | 3705/08/10 | 3720 ram | 3720 cart | |
| Read_Sts %CHi.MOD | | | | |
| Quelque soit le métier excepté la voie de communication du processeur | 30 | 30 | 32 | 2 |
| Read_Sts %CHi | | | | |
| Entrée analogique | 180 | 180 | 216 | 6 |
| Sortie analogique | 90 | 70 | 74 | |
| Module de comptage CTZ | 110 | 95 | 104 | |
| Write_Param %CHi | | | | |
| Entrée analogique | 790 | 570 | 790 | 6 |
| Module de comptage CTZ | 1127 | 1080 | 1083 | |
| Read_Param %CHi | | | | |
| Entrée analogique | 260 | 290 | 316 | 6 |
| Module de comptage CTZ | 338 | 295 | 300 | |
| Save_Param %CHi | | | | |
| Entrée analogique | 1234 | 1220 | 1240 | 6 |
| Module de comptage CTZ | 1370 | 1220 | 1240 | |
| Restore_Param %CHi | | | | |
| Entrée analogique | 550 | 510 | 535 | 6 |
| Module de comptage CTZ | 1160 | 1080 | 1097 | |
| Write_Cmd %CHi | | | | |
| Sortie TOR | 50 | 47 | 52 | 6 |

6.3 performances des instructions sur automates Premium

Présentation

Objet de ce sous chapitre Ce sous chapitre décrit les performances des instructions exécutées sur automate Premium.

Contenu de ce sous-chapitre Ce sous-chapitre contient les sujets suivants :

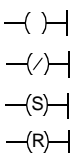
| Sujet | Page |
|---|------|
| Performances des instructions booléennes | 160 |
| Performances des blocs fonction | 164 |
| Arithmétique entière et à virgule flottante | 167 |
| Instructions sur programme et structures de contrôle | 171 |
| Conversions numériques | 173 |
| Instructions sur chaîne de bits | 174 |
| Instruction sur tableaux de mots, doubles mots et flottants | 177 |
| Instructions de gestion du temps | 185 |
| Instructions sur chaînes de caractères | 187 |
| Fonctions métier et fonctions Orphée | 190 |
| Instructions d'entrées/sorties explicite | 193 |
| Bloc fonction DFB | 195 |

Performances des instructions booléennes

Performances des processeurs P57 1•/2•

Le tableau ci-après décrit les performances des instructions booléennes des processeurs P57 1•/2•.

| LD | IL | ST | Objets | Temps d'exécution (µs) | | | | | | | | Taille (mots) |
|-------------|--|----|---------------|------------------------|---------------|-----------------------|------------------------|--------------|---------------|-----------------------|------------------------|---------------|
| | | | | 57 1• ram | 57 1• cart | 57 1• ram > 4 K | 57 1• cart > 4 K | 57 2• ram | 57 2• cart | 57 2• ram > 4 K | 57 2• cart > 4 K | |
| ┌ | | | | 0,37 | 0,50 | | | 0,06 | 0,21 | | | 1 |
| ┌┌ ┌┌┌ | LD LDN | | %M1 (1) | 0,50 | 0,62 | 0,62 | 0,87 | 0,19 | 0,21 | 0,25 | 0,42 | 1 |
| | | | %M1[%MW2] | 1,50 | 2,25 | 1,50 | 2,25 | 0,62 | 1,25 | 0,62 | 1,25 | 6 |
| | | | %MW0:X0 (2) | 1,12 | 1,62 | 1,12 | 1,62 | 0,37 | 0,83 | 0,37 | 0,83 | 4 |
| | | | %IWij:Xk (3) | 1,75 | 2,50 | 1,75 | 2,50 | 0,62 | 1,25 | 0,62 | 1,25 | 6 |
| | | | %MW0[%MW8]:X0 | 2,25 | 3,37 | 2,25 | 3,37 | 0,94 | 1,87 | 0,94 | 1,87 | 9 |
| | | | %KW0[%MW8]:X0 | 2,25 | 3,37 | 2,25 | 3,37 | 0,94 | 1,87 | 0,94 | 1,87 | 9 |
| ┌┌┌ ┌┌┌┌ | LDR, LDF | | %M1 | 0,87 | 1,12 | 1,00 | 1,37 | 0,25 | 0,42 | 0,31 | 0,62 | 2 |
| | | | %M1[%MW2] | 1,87 | 2,75 | 1,87 | 2,75 | 0,69 | 1,46 | 0,69 | 1,46 | 7 |
| ┌┌ ┌┌┌ | AND, ANDN , AND (, AND (N , idem OR | | | idem LD,LDN | | | | | | | | |
| ┌┌┌ ┌┌┌┌ | ANDR, ANDF, AND (R, AND (F, idem OR | | | idem LDR,LDF | | | | | | | | |

| LD | IL | ST | Objets | Temps d'exécution (µs) | | | | | | | | Taille (mots) |
|---|-----------------------|---------|-------------------------|------------------------|---------------|-----------------------|------------------------|--------------|---------------|-----------------------|------------------------|---------------|
| | | | | 57 1• ram | 57 1• cart | 57 1• ram > 4 K | 57 1• cart > 4 K | 57 2• ram | 57 2• cart | 57 2• ram > 4 K | 57 2• cart > 4 K | |
| | XOR XORN | | %M1 | 2,12 | 2,75 | 2,37 | 3,25 | 0,56 | 1,04 | 0,69 | 1,46 | 5 |
| | | | %M1[%MW2] | 3,37 | 4,75 | 3,37 | 4,75 | 0,94 | 2,29 | 0,94 | 2,29 | 12 |
| | | | %MW0:X0 | 3,37 | 4,62 | 3,37 | 4,62 | 0,75 | 2,08 | 0,75 | 2,08 | 10 |
| | | | %IW{i}.j:Xk | 4,00 | 5,50 | 4,00 | 5,50 | 1,00 | 2,50 | 1,00 | 2,50 | 12 |
| | | | %MW0[%MW8]:X0 | 4,50 | 6,37 | 4,50 | 6,37 | 1,31 | 3,12 | 1,31 | 3,12 | 15 |
| | | | %KW0[%MW8]:X0 | 4,50 | 6,37 | 4,50 | 6,37 | 1,31 | 3,12 | 1,31 | 3,12 | 15 |
| | XORR , XORF | | %M1 | 3,62 | 4,75 | 3,87 | 5,25 | 0,81 | 1,87 | 0,94 | 2,29 | 9 |
| | | | %M1[%MW2] | 5,62 | 8,00 | 5,62 | 8,00 | 1,69 | 3,96 | 1,69 | 3,96 | 19 |
|  | ST STN S R | | %M1 (1) | 1,00 | 1,25 | 1,12 | 1,50 | 0,37 | 0,46 | 0,44 | 0,62 | 2 |
| | | | %M1[%MW2] | 1,62 | 2,37 | 1,62 | 2,37 | 0,75 | 1,29 | 0,75 | 1,29 | 6 |
| | | | %MW0:X0 | 1,25 | 1,75 | 1,25 | 1,75 | 0,50 | 0,83 | 0,50 | 0,83 | 4 |
| | | | %NW{i}.j:Xk (3) | 1,87 | 2,62 | 1,87 | 2,62 | 0,75 | 1,25 | 0,75 | 1,25 | 6 |
| | | | %MW0[%MW8]:X0 | 2,37 | 3,50 | 2,37 | 3,50 | 0,75 | 1,92 | 0,75 | 1,92 | 9 |
| bobines multiples en ladder, «coût» de la 2ème bobine et des suivantes | | | | 0,62 | 0,75 | 0,75 | 1,00 | 0,31 | 0,25 | 0,37 | 0,42 | 1 |
| bloc opération | [action] | | bloc exécuté | 0,25 | 0,50 | 0,25 | 0,50 | 0,12 | 0,42 | 0,12 | 0,42 | 2 |
| | | | non exécuté | 0,50 | 0,75 | 0,50 | 0,75 | 0,12 | 0,42 | 0,12 | 0,42 | 2 |
| bloc opération vertical | | | entre 2 %MWi | 1,62 | 11,85 | 1,62 | 2,25 | 0,56 | 1,04 | 0,56 | 1,04 | 5 |
| Converge nce | [)] | [)] | bloc exécuté | 0,37 | 0,13 | 0,37 | 0,50 | 0,06 | 0,21 | 0,06 | 0,21 | 1 |
| Divergenc e non suivie d'une convergen ce | [)] | | Ladder, 1 divergence | 0,37 | 0,50 | 0,37 | 0,50 | 0,06 | 0,21 | 0,06 | 0,21 | 1 |
| | [MPS, MPP, MRD] | | List MPS+MPP | 1,12 | 1,50 | 1,12 | 1,50 | 0,19 | 0,62 | 0,19 | 0,62 | 3 |
| | | | List MRD | 0,37 | 0,50 | 0,37 | 0,50 | 0,06 | 0,21 | 0,06 | 0,21 | 1 |

Légende

1. Ceci concerne tous les objets bits forçables : %I, %Q, %X, %M, %S,
2. autres objets de même type : bits de sortie de bloc fonction %TMi.Q ..., bits extraits de mots système %SWi:Xj
3. autres objets de même type : bits extraits de mots communs %NW{ij}:Xk, bits extraits de mots d'E/S %IWi.j.Xk, %QWi.j.Xk, bits extraits de %KW, bits de défaut %li.j.ERR.

**Performances
des processeurs
P57 3•/4•**

Le tableau ci-après décrit les performances des instructions booléennes des processeurs P57 3•/4•.

| LD | IL | ST | Objets | Temps d'exécution (µs) | | | | | | Taille en mots |
|----|---|----|---------------|------------------------|---------------|-----------------------|------------------------|--------------|---------------|----------------|
| | | | | 57 3• ram | 57 3• cart | 57 3• ram > 4 K | 57 3• cart > 4 K | 57 4• ram | 57 4• > 4K | |
| | | | | 0,04 | 0,17 | | | 0,02 | | 1 |
| | LD LDN | | %M1 (1) | 0,12 | 0,17 | 0,17 | 0,33 | 0,06 | 0,08 | 1 |
| | | | %M1[%MW2] | 0,42 | 1,00 | 0,42 | 1,00 | 0,21 | 0,21 | 6 |
| | | | %MW0:X0 (2) | 0,25 | 0,67 | 0,25 | 0,67 | 0,12 | 0,12 | 4 |
| | | | %IWi.j:Xk (3) | 0,42 | 1,00 | 0,42 | 1,00 | 0,21 | 0,21 | 6 |
| | | | %MW0[%MW8]:X0 | 0,62 | 1,50 | 0,62 | 1,50 | 0,31 | 0,31 | 9 |
| | | | %KW0[%MW8]:X0 | 0,62 | 1,50 | 0,62 | 1,50 | 0,31 | 0,31 | 9 |
| | LDR, LDF | | %M1 | 0,17 | 0,33 | 0,21 | 0,50 | 0,08 | 0,10 | 2 |
| | | | %M1[%MW2] | 0,46 | 1,17 | 0,46 | 1,17 | 0,23 | 0,23 | 7 |
| | AND, ANDN , AND (, AND (N , idem OR | | | idem LD,LDN | | | | | | |
| | | | | idem LD,LDN | | | | | | |
| | ANDR, ANDF, AND (R, AND (F, idem OR | | | idem LD,LDN | | | | | | |
| | | | | idem LD,LDN | | | | | | |

| LD | IL | ST | Objets | Temps d'exécution (µs) | | | | | | Taille en mots |
|--|-----------------------|------|----------------------|------------------------|---------------|-----------------------|------------------------|--------------|---------------|----------------|
| | | | | 57 3• ram | 57 3• cart | 57 3• ram > 4 K | 57 3• cart > 4 K | 57 4• ram | 57 4• > 4K | |
| | XOR XORN | | %M1 | 0,37 | 0,83 | 0,46 | 1,17 | 0,19 | 0,23 | 5 |
| | | | %M1[%MW2] | 0,62 | 1,83 | 0,62 | 1,83 | 0,31 | 0,31 | 12 |
| | | | %MW0:X0 | 0,50 | 1,67 | 0,50 | 1,67 | 0,25 | 0,25 | 10 |
| | | | %IWi.j:Xk | 0,67 | 2,00 | 0,67 | 2,00 | 0,33 | 0,33 | 12 |
| | | | %MW0[%MW8]:X0 | 0,87 | 2,50 | 0,87 | 2,50 | 0,44 | 0,44 | 15 |
| | | | %KW0[%MW8]:X0 | 0,87 | 2,50 | 0,87 | 2,50 | 0,44 | 0,44 | 15 |
| | XORR, XORF | | %M1 | 0,54 | 1,50 | 0,62 | 1,83 | 0,27 | 0,31 | 9 |
| | | | %M1[%MW2] | 1,12 | 3,17 | 1,12 | 3,17 | 0,56 | 0,56 | 19 |
| | ST STN S R | | %M1 (1) | 0,25 | 0,33 | 0,29 | 0,50 | 0,12 | 0,15 | 2 |
| | | | %M1[%MW2] | 0,50 | 1,00 | 0,50 | 1,00 | 0,25 | 0,25 | 6 |
| | | | %MW0:X0 | 0,33 | 0,67 | 0,33 | 0,67 | 0,17 | 0,17 | 4 |
| | | | %NW{i}.j:Xk (3) | 0,50 | 1,00 | 0,50 | 1,00 | 0,25 | 0,25 | 6 |
| | | | %MW0[%MW8]:X0 | 0,62 | 0,75 | 0,50 | 1,50 | 0,25 | 0,25 | 9 |
| bobines multiples en ladder, «coût» de la 2ème bobine et des suivantes | | | | 0,21 | 0,17 | 0,25 | 0,33 | 0,10 | 0,12 | 1 |
| bloc opération | [action] | | bloc exécuté | 0,25 | 0,50 | 0,08 | 0,33 | 0,04 | 0,04 | 2 |
| | | | non exécuté | 0,50 | 0,75 | 0,08 | 0,33 | 0,04 | 0,04 | 2 |
| bloc opérationvertical | | | entre 2 %MWi | 1,62 | 11,85 | 0,37 | 0,83 | 0,19 | 0,19 | 5 |
| Convergence | [)] | [)] | bloc exécuté | 0,37 | 0,13 | 0,04 | 0,17 | 0,02 | 0,02 | 1 |
| Divergence non suivie d'une convergence | [] | | Ladder, 1 divergence | 0,37 | 0,50 | 0,04 | 0,17 | 0,02 | 0,02 | 1 |
| | [MPS, MPP, MRD] | | List MPS+MPP | 1,12 | 1,50 | 0,12 | 0,50 | 0,06 | 0,06 | 3 |
| | | | List MRD | 0,37 | 0,50 | 0,04 | 0,17 | 0,02 | 0,02 | 1 |

Légende

1. Ceci concerne tous les objets bits forçables : %I, %Q, %X, %M, %S,
2. autres objets de même type : bits de sortie de bloc fonction %TMI.Q ..., bits extraits de mots système %SWi:Xj
3. autres objets de même type : bits extraits de mots communs %NW{i}.Xk, bits extraits de mots d'E/S %IWi.j.Xk, %QWi.j.Xk, bits extraits de %KW, bits de défaut %li.j.ERR.

Performances des blocs fonction

Temporisateur IEC Le tableau ci-après décrit les performances du bloc fonction temporisateur IEC

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | | | Taille (mots) |
|-------------------------|----------------------------|------------|-----------------|------------------------|------|------|------|---------------|
| | | | | 571• | 572• | 573• | 574• | |
| front montant sur IN | IN %TM1 (front montant) | START %TM1 | lancement timer | 29 | 8,0 | 5,4 | 3,7 | 3 |
| front descendant sur IN | IN %TM1 (front descendant) | DOWN %TM1 | arrêt timer | 9 | 2,6 | 1,7 | 1,2 | |
| IN =1 | IN %TM1 (=1) | | timer actif | 12 | 3,5 | 2,3 | 1,6 | |
| IN =0 | IN %TM1 (=0) | | timer inactif | 10 | 3,3 | 2,2 | 1,5 | |

Temporisateur PL7-3 Le tableau ci-après décrit les performances du bloc fonction temporisateur PL7-3

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | | | Taille (mots) |
|------|----|-----------|---------------|------------------------|------|------|------|---------------|
| | | | | 571• | 572• | 573• | 574• | |
| | | START %T1 | armer | | | | | 3 |
| | | STOP %T1 | geler | 7 | 2,8 | 2,0 | 1,4 | |
| E =0 | | RESET %T1 | reset | 7 | 3,1 | 2,2 | 1,6 | |
| | | | timer actif | 11 | 3,4 | 2,3 | 1,7 | |
| | | | timer inactif | | | | | |

**Compteur/
décompteur**

Le tableau ci-après décrit les performances du bloc fonction compteur/décompteur

| LD | IL | ST | Condition s | Temps d'exécution (µs) | | | | Taille (mots) |
|----------------------------|--------------------------|---------------|------------------|------------------------|------|------|------|------------------|
| | | | | 571• | 572• | 573• | 574• | |
| reset, R=1 | R %C8 (=1) | RESET %C8 | reset | 11 | 3,4 | 2,3 | 1,7 | 3 |
| preset, S=1 | S %C9 (=1) | PRESET %C9 | preset | 12 | 3,6 | 2,4 | 1,7 | |
| front montant sur CU | CU %C8 (front) | UP %C8 | up | 12 | 3,7 | 2,5 | 1,8 | |
| front montant sur CD | CD %C9 (front) | DOWN %C9 | down | 12 | 3,7 | 2,5 | 1,8 | |
| entrées inactives | R/S/CU/CD bit inactif | | aucune action | 7 | 2,5 | 1,7 | 1,2 | |

Monostable

Le tableau ci-après décrit les performances du bloc fonction monostable

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | | | Taille (mots) |
|---------------------------|-----------------------------|---------------|---------------------|------------------------|------|------|------|------------------|
| | | | | 571• | 572• | 573• | 574• | |
| front montant sur S | S %MN0, front montant | START %MN0 | lancement | 24 | 7,2 | 4,9 | 3,4 | 3 |
| S=1 | S %MN0, S =1/0 | | monostable actif | 6 | 2,2 | 1,5 | 1,1 | |

Registre

Le tableau ci-après décrit les performances du bloc fonction registre

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | | | Taille (mots) |
|-------------------|--------------------|-----------|--------------|------------------------|------|------|------|---------------|
| | | | | 571• | 572• | 573• | 574• | |
| front sur I | I %R2 (front) | PUT %R2 | stockage | 13 | 3,9 | 2,6 | 1,8 | 3 |
| front sur O | O %R2 (front) | GET %R2 | déstockage | 13 | 3,9 | 2,6 | 1,8 | |
| R=1 | R %R1 (=1) | RESET %R2 | reset | 9 | 3,3 | 2,3 | 1,6 | |
| entrées inactives | I/O/R, bit inactif | | pas d'action | 6 | 2,6 | 1,8 | 1,3 | |

Programmeur cyclique

Le tableau ci-après décrit les performances du programmeur cyclique

| LD | IL | ST | Conditions | Temps d'exécution (µs) | | | | Taille (mots) |
|-------------------|------------------|------------|--------------------|------------------------|------|------|------|---------------|
| | | | | 571• | 572• | 573• | 574• | |
| front sur U | U %DR0 | UP %DR1 | up, fixe | 124 | 35 | 24 | 16 | 3 |
| | | | par bit d'ordre | 25 | 25 | 25 | 25 | |
| R=1 | R %DR1 | RESET %DR2 | reset, fixe | 118 | 33 | 23 | 15 | |
| | | | par bit d'ordre | 25 | 25 | 25 | 25 | |
| entrées inactives | R/U, bit inactif | | pas d'action, fixe | 120 | 34 | 23 | 16 | |
| | | | par bit d'ordre | 25 | 25 | 25 | 25 | |

Arithmétique entière et à virgule flottante

Corrections en fonction du type d'objets

Les temps et volumes pages ci-après sont donnés pour des objets de type %MW0, %MD0 ou %MF0.

Le tableau suivant décrit les corrections à apporter aux valeurs données dans le tableau de performances des instructions arithmétiques si d'autres types d'objets sont utilisés.

| Type d'objet | Position de l'objet | Type de correction | Temps d'exécution (μ s) | | | | | | | Taille (mots) |
|--|--|--|------------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| Valeur immédiate simple longueur | - | Valeur à enlever ou ajouter à celle mentionnée pour %MW | -0,12 | -0,12 | -0,12 | 0,00 | -0,08 | 0,00 | -0,04 | 0 |
| Valeur immédiate double longueur | - | Valeur à enlever ou ajouter à celle mentionnée pour %MD ou %MF | -0,12 | 0,00 | -0,19 | 0,21 | -0,12 | 0,17 | -0,06 | 1 |
| Mots, doubles Mots ou flottants indexés | Après le signe := | Valeur à ajouter | 1,12 | 1,75 | 0,56 | 1,04 | 0,37 | 0,83 | 0,19 | 5 |
| | Première opération, le premier opérande n'étant pas indexé, ou affectation | Valeur à ajouter | 1,12 | 1,75 | 0,56 | 1,04 | 0,37 | 0,83 | 0,19 | 5 |
| | Deuxième opérande si le premier opérande est également indexé | Valeur à ajouter | 1,12 | 1,75 | 0,56 | 1,04 | 0,37 | 0,83 | 0,19 | 5 |

| Type d'objet | Position de l'objet | Type de correction | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--|---------------------|--------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %KWi, %KWj[%MWj] %KDi, %KFi mots commun, mots d'entrées/ sorties | - | Valeur à ajouter | 0,62 | 0,87 | 0,25 | 0,42 | 0,17 | 0,33 | 0,08 | 2 |

Correction en fonction du contexte de l'opération

Le tableau suivant décrit les corrections à apporter aux valeurs données dans le tableau de performances des instructions arithmétiques en fonction du contexte de l'opération.

| Contexte de l'opération | Type d'objet | Type de correction | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--|--------------|---|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| L'opération est au minimum en deuxième position dans la phrase Ex : %MW2 dans :=%MW0*%MW1*%MW2 | %MW | Valeur à ajouter à celle mentionnée pour %MW | 0,50 | 0,62 | 0,19 | 0,33 | 0,12 | 0,25 | 0,06 | 1 |
| | %MD ou %MF | Valeur à ajouter à celle mentionnée pour %MD ou %MF | 0,62 | 0,75 | 0,31 | 0,46 | 0,21 | 0,33 | 0,10 | 1 |
| Opération avec résultat d'une opération entre parenthèse ou plus prioritaire Ex : %MW0+%MW2+(...) | %MW | Valeur à ajouter à celle mentionnée pour %MW | 0,37 | 0,37 | 0,12 | 0,12 | 0,08 | 0,08 | 0,04 | 1 |
| | %MD ou %MF | Valeur à ajouter à celle mentionnée pour %MD ou %MF | 0,50 | 0,50 | 0,25 | 0,25 | 0,17 | 0,17 | 0,08 | 1 |

Tableau de performance des instructions Le tableau ci-après décrit les performances des instructions arithmétiques.

| ST | Objets | Conditions | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---------------------|--------------|-------------------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| objet après le := | %MW0 | - | 0,6 | 0,9 | 0,2 | 0,4 | 0,2 | 0,3 | 0,08 | 2 |
| | | %MW0+(...ou %MW0 avant *, / ou REM) | 0,7 | 1,0 | 0,4 | 0,4 | 0,2 | 0,3 | 0,12 | 2 |
| | %MD0 | - | 0,7 | 1,0 | 0,4 | 0,4 | 0,2 | 0,3 | 0,12 | 2 |
| | | %MW0+(...ou %MW0 avant *, / ou REM) | 1,0 | 1,2 | 0,6 | 0,5 | 0,4 | 0,3 | 0,21 | 2 |
| %MF0 | | 1,0 | 1,2 | 0,6 | 0,5 | 0,4 | 0,3 | 0,21 | 2 | |
| := | %MW0 | | 0,6 | 0,9 | 0,2 | 0,4 | 0,2 | 0,3 | 0,08 | 2 |
| | %MD0 et %MF0 | | 0,7 | 1,0 | 0,4 | 0,4 | 0,2 | 0,3 | 0,12 | 2 |
| =, <>, <=, <, >, >= | %MW0 | | 1,0 | 1,4 | 0,3 | 0,6 | 0,2 | 0,5 | 0,10 | 3 |
| | %MD0 | | 1,5 | 1,5 | 0,4 | 0,6 | 0,3 | 0,5 | 0,15 | 3 |
| | %MF0 | | 24 | 24 | 2,6 | 2,6 | 1,9 | 2,0 | 1,4 | 4 |
| AND, OR, XOR | %MW0 | | 0,6 | 0,9 | 0,2 | 0,4 | 0,2 | 0,3 | 0,08 | 2 |
| | %MD0 | | 0,7 | 1,0 | 0,4 | 0,4 | 0,2 | 0,3 | 0,12 | 2 |
| +, - | %MW0 | | 0,6 | 0,9 | 0,2 | 0,4 | 0,2 | 0,3 | 0,08 | 2 |
| | %MD0 | | 0,7 | 1,0 | 0,4 | 0,4 | 0,2 | 0,3 | 0,12 | 2 |
| | %MF0 | | 48 | 48 | 2,5 | 2,5 | 1,9 | 2,0 | 1,4 | 3 |
| * | %MW0 | | 6,3 | 6,5 | 2,0 | 2,2 | 1,5 | 1,6 | 1,2 | 3 |
| | %MD0 | | 29 | 29 | 9,3 | 9,3 | 6,3 | 6,4 | 4,7 | 3 |
| | %MF0 | | 44 | 44 | 2,5 | 2,5 | 1,9 | 2,0 | 1,4 | 3 |
| /, REM | %MW0 | | 6,9 | 7,2 | 2,1 | 2,3 | 1,5 | 1,6 | 1,2 | 3 |
| | %MD0 | | 148 | 149 | 34 | 34 | 21 | 21 | 15 | 3 |
| / | %MF0 | | 46 | 47 | 3,3 | 3,4 | 2,5 | 2,6 | 1,8 | 3 |
| ABS, -objet | %MW0 | | 0,6 | 0,9 | 0,2 | 0,4 | 0,2 | 0,3 | 0,08 | 2 |
| | %MD0 | | 0,7 | 1,0 | 0,4 | 0,4 | 0,2 | 0,3 | 0,12 | 2 |
| | %MF0 | | 9 | 10 | 2,1 | 2,1 | 1,6 | 1,7 | 1,2 | 3 |
| NOT | %MW0 | | 0,6 | 0,9 | 0,2 | 0,4 | 0,2 | 0,3 | 0,1 | 2 |
| | %MD0 | | 0,7 | 1,0 | 0,4 | 0,4 | 0,2 | 0,3 | 0,1 | 2 |

| ST | Objets | Conditions | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-----------------------|--------|---------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| SQRT | %MW0 | | 19 | 19 | 3,5 | 3,7 | 2,1 | 2,2 | 1,5 | 3 |
| | %MD0 | | 62 | 62 | 10,2 | 10,3 | 5,7 | 5,8 | 4,4 | 3 |
| | %MF0 | | 117 | 117 | 2,8 | 2,8 | 2,1 | 2,1 | 1,5 | 3 |
| INC, DEC | %MW0 | | 0,7 | 1,0 | 0,4 | 0,4 | 0,2 | 0,3 | 0,12 | 2 |
| | %MD0 | | 1,0 | 1,2 | 0,6 | 0,5 | 0,4 | 0,3 | 0,21 | 2 |
| SHL, SHR, ROL, ROR | %MW0 | pour 1 bit | 2,0 | 2,9 | 0,8 | 1,5 | 0,5 | 1,2 | 0,27 | 7 |
| | %MD0 | pour 1 bit | 2,1 | 3,0 | 0,9 | 1,5 | 0,6 | 1,2 | 0,31 | 7 |
| | | par bit supplémentaire | 0,042 | | | | | | | |
| LN | %MF0 | | 847 | 847 | 2,2 | 2,2 | 1,6 | 1,6 | 1,5 | |
| LOG | %MF0 | | 900 | 900 | 2,2 | 2,2 | 1,6 | 1,6 | 1,5 | |
| EXP | %MF0 | | 713 | 713 | 6,4 | 6,4 | 4,7 | 4,7 | 4,0 | |
| EXPT | %MF0 | | 1 747 | 1 747 | 2,2 | 2,2 | 1,6 | 1,6 | 1,5 | |
| TRUNC | %MF0 | | 1 753 | 1 753 | 2,2 | 2,2 | 1,6 | 1,6 | 1,5 | |
| COS | %MF0 | | 1 813 | 1 813 | 2,2 | 2,2 | 1,6 | 1,6 | 1,5 | |
| SIN | %MF0 | | 2 520 | 2 520 | 2,7 | 2,7 | 2,0 | 2,0 | 1,8 | |
| TAN | %MF0 | | 2 520 | 2 520 | 3,4 | 3,4 | 2,5 | 2,5 | 2,2 | |
| ACOS | %MF0 | | 1 720 | 1 720 | 2,1 | 2,1 | 1,6 | 1,6 | 1,4 | |
| ASIN | %MF0 | | 1 640 | 1 640 | 61 | 68 | 43 | 49 | 32 | |
| ATAN | %MF0 | | 103 | 142 | 32 | 36 | 23 | 26 | 17 | |
| DEG_TO_RA D | %MF0 | | 392 | 537 | 86 | 96 | 61 | 69 | 45 | |
| RAD_TO_DE G | %MF0 | | 380 | 522 | 86 | 96 | 61 | 69 | 46 | |

Instructions sur programme et structures de contrôle

Performance des instructions sur programme Le tableau ci-après décrit les performances des instructions sur programme.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-----------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• |
| Jump %Li | 1,2 | 1,6 | 0,8 | 0,6 | 0,5 | 0,5 | 0,3 | 3 |
| Maskevt | 15,5 | 15,5 | 5,8 | 5,8 | 4,0 | 4,0 | 2,8 | 1 |
| Unmaskevt | 15,7 | 15,7 | 6,0 | 6,0 | 4,2 | 4,2 | 3,0 | 1 |
| SRi | 1,9 | 2,2 | 1,4 | 1,2 | 1,0 | 0,8 | 0,5 | 2 |
| Return | 0,6 | 0,9 | 0,2 | 0,4 | 0,2 | 0,3 | 0,1 | 2 |

Performance des Structures de contrôle Le tableau ci-après décrit les performances des instructions de type structures de contrôle.

| ST | | Temps d'exécution (µs) | | | | | | | Volume (mots) |
|---|---|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• |
| <cond> | évaluation de la condition | | | | | | | | |
| bit forçable | voir instruction booléenne LD %M1 | | | | | | | | |
| comparaison | voir comparaisons =,<,> ... | | | | | | | | |
| if <cond > then <action> end_if; | les temps et volumes indiqués ci-dessous sont à ajouter à ceux de l'action contenue dans la structure | | | | | | | | |
| condition vraie | | 0,2 | 0,5 | 0,1 | 0,4 | 0,1 | 0,3 | 0,04 | 2 |
| condition fausse (saut) | | 0,5 | 0,7 | 0,1 | 0,4 | 0,1 | 0,3 | 0,04 | |
| If <cond> then <action1> else <action2> end_if; | | | | | | | | | |
| condition vraie | | 0,7 | 1,2 | 0,2 | 0,8 | 0,2 | 0,7 | 0,08 | 4 |
| condition fausse | | 0,5 | 0,7 | 0,1 | 0,4 | 0,1 | 0,3 | 0,04 | |
| while <cond> do.<action> end_while | | | | | | | | | |
| passage dans la boucle avec rebouclage | | 0,7 | 1,2 | 0,2 | 0,8 | 0,2 | 0,7 | 0,08 | 2 |
| sortie de la boucle | | 0,5 | 0,7 | 0,1 | 0,4 | 0,1 | 0,3 | 0,04 | |
| repeat <action> until <cond> end_repeat | | | | | | | | | |

| ST | | Temps d'exécution (μs) | | | | | | | Volume (mots) |
|--|--|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• |
| passage dans la boucle avec rebouclage | | 0,5 | 0,7 | 0,1 | 0,4 | 0,1 | 0,3 | 0,04 | 2 |
| dernier passage | | 0,2 | 0,5 | 0,1 | 0,4 | 0,1 | 0,3 | 0,04 | |
| for <mot1:=mot2>to <mot3> do <action> end_for | | | | | | | | | |
| entrée dans le for, exécuté une seule fois | | 1,2 | 1,7 | 0,5 | 0,8 | 0,3 | 0,7 | 0,17 | 15 |
| passage dans la boucle avec rebouclage | | 3,5 | 5,0 | 1,2 | 2,5 | 0,8 | 2,0 | 0,42 | |
| sortie de la boucle | | 1,7 | 2,5 | 0,6 | 1,2 | 0,4 | 1,0 | 0,21 | |

Conversions numériques

Performances Le tableau ci-après décrit les performances des instructions de conversions numériques.

| ST | Temps d'exécution (µs) | | | | | | | Volume (mots) |
|--------------------|------------------------|------------|-----------|------------|-----------|------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| BCD_TO_INT(%MW0) | 17 | 17 | 3,3 | 3,4 | 1,9 | 2,1 | 1,5 | 3,0 |
| INT_TO_BCD(%MW0) | 14 | 14 | 2,8 | 3,0 | 1,7 | 1,9 | 1,3 | 3,0 |
| GRAY_TO_INT(%MW0) | 27 | 28 | 4,7 | 4,9 | 2,7 | 2,8 | 1,9 | 3,0 |
| INT_TO_REAL(%MW0) | 28 | 28 | 1,5 | 1,7 | 1,4 | 1,6 | 1,0 | 3,0 |
| DINT_TO_REAL(%MD0) | 24 | 24 | 1,7 | 1,8 | 1,6 | 1,7 | 1,2 | 3,0 |
| REAL_TO_INT(%MF0) | 41 | 42 | 1,6 | 1,7 | 1,5 | 1,6 | 1,1 | 3,0 |
| REAL_TO_DINT(%MF0) | 33 | 33 | 1,7 | 1,8 | 1,6 | 1,7 | 1,2 | 3,0 |
| DBCD_TO_DINT(%MD0) | 612 | 840 | 231 | 233 | 178 | 179 | 138 | 5 |
| DBCD_TO_INT(%MD0) | 537 | 737 | 203 | 204 | 156 | 157 | 121 | 5 |
| DINT_TO_DBCD(%MD0) | 512 | 702 | 193 | 195 | 149 | 150 | 115 | 5 |
| INT_TO_DBCD(%MW0) | 274 | 376 | 104 | 104 | 80 | 80 | 62 | 5 |

Instructions sur chaîne de bits

Initialisation d'un tableau de bits Le tableau ci-après décrit les performances des instructions d'initialisation d'un tableau de bits.

| ST | Taille (bit) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--------------|--------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %M30:8 := 0 | 8 | 3,6 | 4,5 | 2,4 | 3,2 | 1,6 | 2,3 | 0,8 | 7 |
| %M30:16 := 1 | 16 | 5,6 | 6,5 | 4,4 | 5,2 | 3,0 | 3,7 | 1,5 | 7 |
| %M30:24 := 2 | 24 | 12 | 14 | 5,6 | 6,8 | 3,7 | 4,8 | 2,4 | 12 |
| %M30:32 := 2 | 32 | 14 | 16 | 7,6 | 8,8 | 5,1 | 6,1 | 3,1 | 12 |

Copie d'un tableau de bits dans un tableau de bits Le tableau ci-après décrit les performances des instructions de copie d'un tableau de bits dans un autre tableau de bits.

| ST | Taille (bit) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---------------------------------|--------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %M30:8 := %M20:8 | 8 | 6,9 | 7,9 | 5,4 | 6,0 | 3,6 | 4,2 | 1,8 | 8 |
| %M30:16 := %M20:16 | 16 | 8,1 | 9,1 | 6,6 | 7,3 | 4,4 | 5,1 | 2,2 | 8 |
| %M30:24 := %M20:24 | 24 | 22 | 23 | 14 | 16 | 10 | 11 | 5,4 | 13 |
| %M30:32 := %M20:32 | 32 | 27 | 28 | 19 | 21 | 13 | 14 | 7,0 | 13 |
| %M30:16 := COPY_BIT(%M20:16) | 16 | 173 | 237 | 65 | 66 | 50 | 50 | 39 | 17 |
| | 32 | 263 | 360 | 99 | 100 | 76 | 77 | 59 | 17 |
| | 128 | 818 | 1 122 | 309 | 312 | 238 | 239 | 184 | 17 |

Instructions logiques sur tableau de bits

Le tableau ci-après décrit les performances des instructions logiques sur tableau de bits.

| ST | Taille (bits) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--|---------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| AND_ARX, OR_ARX, XOR_ARX | | | | | | | | | |
| %M0:16 := AND_ARX(%M30:16,%M50:16) | 16 | 233 | 319 | 88 | 89 | 68 | 68 | 52 | 24 |
| %M0:32 := AND_ARX(%M30:32,%M50:32) | 32 | 368 | 504 | 139 | 140 | 107 | 107 | 83 | 24 |
| %M0:128 := AND_ARX(%M30:128,%M50:128) | 128 | 1 178 | 1 616 | 445 | 449 | 343 | 344 | 265 | 24 |
| NOT_ARX | | | | | | | | | |
| %M0:16 := NOT_ARX(%M30:16) | 16 | 173 | 237 | 65 | 66 | 50 | 50 | 39 | 17 |
| | 32 | 263 | 360 | 99 | 100 | 76 | 77 | 59 | 17 |
| | 128 | 818 | 1 122 | 309 | 312 | 238 | 239 | 184 | 17 |

Copie d'un tableau de bits dans un tableau de mots

Le tableau ci-après décrit les performances des instructions de copie de tableaux de bits dans un tableau de mots.

| ST | Taille (bits) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|------------------------------------|---------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MW1 := %M30:8 | 8 | 4,5 | 5,2 | 3,4 | 3,9 | 2,2 | 2,7 | 1,1 | 6 |
| %MW1 := %M30:16 | 16 | 7,5 | 8,2 | 6,4 | 6,9 | 4,2 | 4,7 | 2,1 | 6 |
| %MD2 := %M30:24 | 24 | 11 | 11 | 10 | 10 | 6,8 | 7,2 | 3,8 | 6 |
| %MD2 := %M30:32 | 32 | 14 | 14 | 13 | 13 | 8,8 | 9,2 | 4,8 | 6 |
| %MW1:4 := BIT_W(%M40:80,0,17,2) | 17 | 231 | 317 | 87 | 88 | 67 | 68 | 52 | 23 |
| %MD1:4 := BIT_D(%M30:80,0,33,0) | 33 | 325 | 446 | 123 | 124 | 95 | 95 | 73 | 23 |

**Copie d'un
tableau de mots
dans un tableau
de bits**

Le tableau ci-après décrit les performances des instructions de copie de tableaux de mots dans un tableau de bits.

| ST | Taille (bits) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-------------------------------------|------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|------------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %M30:8 := %MW1 | 8 | 3,6 | 4,4 | 2,5 | 3,0 | 1,7 | 2,2 | 0,8 | 6 |
| %M30:16 := %MW2 | 16 | 5,6 | 6,4 | 4,5 | 5,0 | 3,0 | 3,5 | 1,5 | 6 |
| %M30:24 := %MD1 | 24 | 12 | 13 | 6,1 | 6,9 | 4,3 | 5,1 | 3,1 | 11 |
| %M30:32 := %MD3 | 32 | 14 | 15 | 8,1 | 8,9 | 5,7 | 6,4 | 3,7 | 11 |
| %M30:32 := W_BIT(%MW200:2,0,2,0) | 32 | 231 | 317 | 87 | 88 | 67 | 68 | 52 | 23 |
| %M30:32 := D_BIT(%MD0:1,0,2,0) | 32 | 275 | 377 | 104 | 105 | 80 | 80 | 62 | 23 |

Instruction sur tableaux de mots, doubles mots et flottants

Initialisation d'un tableau de mots par un mot

Le tableau ci-après décrit les performances des instructions d'initialisation d'un tableau de mots par un mot..

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-------------------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MWO:10 := %MW100 | 10 mots | 34 | 35 | 14 | 15 | 10 | 11 | 6,7 | 10 |
| | par mot | 0,16 | 0,16 | 0,15 | 0,15 | 0,12 | 0,12 | 0,08 | |
| %MD0:10 := %MD100 | 10 double mots | 53 | 54 | 19 | 20 | 13 | 14 | 8,8 | 10 |
| | par double mot | 1,98 | 1,98 | 0,57 | 0,57 | 0,37 | 0,37 | 0,26 | |

Copie d'un tableau de mots dans un tableau de mots

Le tableau ci-après décrit les performances des instructions de copie d'un tableau de mots dans un autre tableau de mots.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--------------------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MWO:10:=%MW20:10; | 10 mots | 63 | 65 | 25 | 26 | 17 | 19 | 12 | 15 |
| | par mot | 0,28 | 0,28 | 0,31 | 0,31 | 0,24 | 0,24 | 0,15 | |
| %MD0:10:=%MD20:10; | 10 double mots | 69 | 71 | 29 | 30 | 20 | 22 | 14 | 15 |
| | par double mot | 0,79 | 0,79 | 0,71 | 0,71 | 0,53 | 0,53 | 0,36 | |

Instructions arithmétiques et logiques entre 2 tableaux de mots

Le tableau ci-après décrit les performances des instructions arithmétiques et logiques entre 2 tableaux de mots.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|------------------------------------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| +, - | | | | | | | | | |
| %MW0:10 := %MW10:10 + %MW20:10; | 10 mots | 110 | 112 | 41 | 43 | 28 | 30 | 19 | 23 |
| | par mot | 4,7 | 4,7 | 1,8 | 1,8 | 1,3 | 1,3 | 0,9 | |
| %MD0:10:= %MD10:10+%MD20:10; | 10 double mots | 154 | 156 | 60 | 62 | 41 | 43 | 28 | 23 |
| | par double mot | 8,9 | 8,9 | 3,6 | 3,6 | 2,5 | 2,5 | 1,7 | |
| * | | | | | | | | | |
| %MW0:10:= %MW10:10 * %MW20:10; | 10 mots | 127 | 129 | 47 | 50 | 33 | 35 | 23 | 23 |
| | par mot | 6,4 | 6,4 | 2,4 | 2,4 | 1,7 | 1,7 | 1,2 | |
| %MD0:10:= %MD10:10 * %MD20:10; | 10 double mots | 441 | 444 | 153 | 155 | 104 | 106 | 73 | 23 |
| | par double mot | 37,6 | 37,6 | 12,9 | 12,9 | 8,7 | 8,7 | 6,2 | |
| /, REM | | | | | | | | | |
| %MW0:10:= %MW10:10 / %MW20:10; | 10 mots | 133 | 135 | 49 | 52 | 34 | 36 | 24 | 23 |
| | par mot | 7,0 | 7,0 | 2,6 | 2,6 | 1,8 | 1,8 | 1,4 | |
| %MD0:10:= %MD10:10 / %MD20:10; | 10 double mots | 1 639 | 1 642 | 395 | 397 | 248 | 250 | 172 | 23 |
| | par double mot | 157 | 157 | 37 | 37 | 23 | 23 | 16 | |
| AND, OR, XOR | | | | | | | | | |
| %MW0:10:=%MW10:10 AND %MW20:10; | 10 mots | 108 | 111 | 40 | 43 | 28 | 30 | 19 | 23 |
| | par mot | 4,5 | 4,5 | 1,7 | 1,7 | 1,2 | 1,2 | 0,8 | |
| %MD0:10:=%MD10:10 AND %MD20:10; | 10 double mots | 155 | 158 | 61 | 63 | 42 | 44 | 29 | 23 |
| | par double mot | 9 | 9 | 4 | 4 | 3 | 3 | 2 | |

Instructions arithmétiques et logiques entre 1 tableau de mots et 1 mot

Le tableau ci-après décrit les performances des instructions arithmétiques et logiques entre 1 tableau de mots et 1 mot.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) | |
|---|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|--|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• | |
| +, - | | | | | | | | | | |
| %MW0:10 :=%MW10:10 + %MW20; ou %MW0:10 := %MW20 + %MW10:10 | 10 mots | 86 | 88 | 30 | 32 | 21 | 22 | 14 | 18 | |
| | par mot | 2,8 | 2,8 | 0,8 | 0,8 | 0,5 | 0,5 | 0,4 | | |
| %MD0:10 :=%MD10:10 + %MD20; | 10 double mots | 112 | 114 | 41 | 43 | 28 | 30 | 19 | 18 | |
| | par double mot | 5,2 | 5,2 | 1,9 | 1,9 | 1,3 | 1,3 | 0,9 | | |
| * | | | | | | | | | | |
| %MW0:10 := %MW20*%MW10:10; | 10 mots | 113 | 115 | 38 | 40 | 26 | 27 | 18 | 18 | |
| | par mot | 5,6 | 5,6 | 1,6 | 1,6 | 1,1 | 1,1 | 0,7 | | |
| %MD0:10:= %MD20*%MD10:10; | 10 double mots | 381 | 383 | 132 | 134 | 90 | 92 | 64 | 18 | |
| | par double mot | 32 | 32 | 11 | 11 | 7,4 | 7,4 | 5,3 | | |
| /, REM | | | | | | | | | | |
| %MW0:10 :=%MW10:10 / %MW30; | 10 mots | 140 | 142 | 46 | 48 | 31 | 33 | 21 | 18 | |
| | par mot | 8,4 | 8,4 | 2,4 | 2,4 | 1,6 | 1,6 | 1,1 | | |
| %MD0:10:= MD10:10 / %MD30 | 10 double mots | 1 585 | 1 587 | 375 | 377 | 235 | 236 | 163 | 18 | |
| | par double mot | 152 | 152 | 35 | 35 | 22 | 22 | 15 | | |
| AND, OR, XOR | | | | | | | | | | |
| %MW0:10 :=%MW10:10 AND %MW20; | 10 mots | 86 | 88 | 30 | 32 | 21 | 22 | 14 | 18 | |
| | par mot | 2,8 | 2,8 | 0,8 | 0,8 | 0,5 | 0,5 | 0,4 | | |

| ST | Taille (du tableau de mot) | Temps d'exécution (μs) | | | | | | | Taille (mots) |
|-------------------------------|----------------------------|------------------------|------------|-----------|------------|-----------|------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MD0:10 :=%MD20 AND %MD10:10; | 10 double mots | 381 | 383 | 132 | 134 | 90 | 92 | 64 | 18 |
| | par double mot | 4,2 | 4,2 | 1,5 | 1,5 | 1,0 | 1,0 | 0,7 | |
| NOT | | | | | | | | | |
| %MW0:10 :=NOT(%MW10:10); | 10 mots | 74 | 75 | 26 | 28 | 18 | 19 | 12 | 15 |
| | par mot | 1,9 | 1,9 | 0,5 | 0,5 | 0,4 | 0,4 | 0,2 | |
| %MD0:10:=NOT(%MD10:10) | 10 double mots | 84 | 86 | 31 | 33 | 22 | 23 | 15 | 15 |
| | par double mot | 2,9 | 2,9 | 1,0 | 1,0 | 0,7 | 0,7 | 0,5 | |

Fonction sommation sur tableau

Le tableau ci-après décrit les performances des instructions de sommation sur tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (μs) | | | | | | | Taille (mots) |
|----------------------|----------------------------|------------------------|------------|-----------|------------|-----------|------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MW20:=SUM(%MW0:10); | 10 mots | 51 | 53 | 17 | 18 | 12 | 12 | 8 | 16 |
| | par mot | 1,6 | 1,6 | 0,4 | 0,4 | 0,3 | 0,3 | 0,2 | |
| %MD20:=SUM(%MD0:10); | 10 double mots | 58 | 59 | 19 | 20 | 13 | 14 | 9 | 16 |
| | par double mot | 2,1 | 2,1 | 0,6 | 0,6 | 0,4 | 0,4 | 0,3 | |

Fonction de comparaison de tableaux Le tableau ci-après décrit les performances des instructions de comparaison de tableaux.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-------------------------------------|----------------------------|------------------------|------------|-----------|------------|-----------|------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MW20:= EQUAL(%MW0:10;%MW10:10); | 10 mots | 67 | 69 | 26 | 28 | 18 | 20 | 13 | 27 |
| | par mot | 0,6 | 0,6 | 0,4 | 0,4 | 0,3 | 0,3 | 0,2 | |
| %MD20:= EQUAL(%MD0:10;%MD10:10); | 10 double mots | 74 | 76 | 31 | 33 | 22 | 23 | 15 | 27 |
| | par double mot | 1,2 | 1,2 | 0,9 | 0,9 | 0,7 | 0,7 | 0,5 | |

Fonction de recherche Le tableau ci-après décrit les performances des instructions de recherche dans un tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-------------------------------------|----------------------------|------------------------|------------|-----------|------------|-----------|------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MW20 := FIND_EQW(%MW0:10,%KW0) | 10 mots, cas max | 150 | 206 | 57 | 57 | 44 | 44 | 34 | 14 |
| %MD20 := FIND_EQD(%MD0:10, %KD0) | 10 double mots, cas max | 163 | 223 | 61 | 62 | 47 | 48 | 37 | 15 |

Recherche de valeurs maxi et mini

Le tableau ci-après décrit les performances des instructions de recherche de valeurs maxi et mini dans un tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|------------------------------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MW20 := MAX_ARW(%MW0:10) | 10 mots | 163 | 223 | 61 | 62 | 47 | 48 | 37 | 12 |
| %MD20 := MAX_ARD(%MD0:10) | 10 double mots | 194 | 266 | 73 | 74 | 56 | 57 | 44 | 12 |

Calcul du nombre d'occurrences

Le tableau ci-après décrit les performances des instructions du nombre d'occurrence d'une valeur dans un tableau de mots.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--------------------------------------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MW20 := OCCUR_ARW(%MW0:10, %KW0) | 10 mots | 163 | 223 | 61 | 62 | 47 | 48 | 37 | 14 |
| %MD20 := OCCUR_ARD(%MD0:10, %KD0) | 10 double mots | 175 | 240 | 66 | 67 | 51 | 51 | 39 | 15 |

Décalage circulaire

Le tableau ci-après décrit les performances des instructions de décalage circulaire.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--------------------------------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| ROL_ARW(mot ou valeur,%MWj:10) | 10 mots | 250 | 343 | 94 | 95 | 73 | 73 | 56 | 12 |
| ROL_ARD(%MDi,%MDj:10) | 10 double mots | 269 | 369 | 102 | 102 | 78 | 79 | 61 | 12 |

Instruction de tri Le tableau ci-après décrit les performances des instructions de tri des éléments d'un tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (μs) | | | | | | | Taille (mots) | |
|------------------------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|--|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• | |
| SORT_ARW(%MWi,%MWj:10) | 10 mots, cas max | 450 | 618 | 170 | 172 | 131 | 132 | 101 | 12 | |
| SORT_ARD(%MDi,%MDj:10) | 5 double mots, cas max | 275 | 377 | 104 | 105 | 80 | 80 | 62 | 12 | |

Calcul de longueur Le tableau ci-après décrit les performances des instructions de calcul de longueur d'un tableau.

| ST | Taille (du tableau de mot) | Temps d'exécution (μs) | | | | | | | Taille (mots) | |
|----------------------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|--|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• | |
| LENGTH_ARW(tab_mot) | | 31 | 43 | 12 | 12 | 9 | 9 | 7 | 6 | |
| LENGTH_ARD(tab_dmot) | | 31 | 43 | 12 | 12 | 9 | 9 | 7 | 6 | |
| LENGTH_ARW(tab_reel) | | 31 | 43 | 12 | 12 | 9 | 9 | 7 | 6 | |
| LENGTH_ARW(tab_bit) | | 31 | 43 | 12 | 12 | 9 | 9 | 7 | 6 | |

Tableaux de flottant

Le tableau ci-après décrit les performances des instructions sur tableau de flottant.

| ST | Taille (du tableau de mot) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-----------|----------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| SUM_ARR | 10 réels | 794 | 942 | 186 | 207 | 132 | 149 | 98 | |
| FIND_EQR | 10 réels, cas médian | 390 | 535 | 83 | 93 | 59 | 67 | 44 | |
| FIND_EQRP | 10 réels, cas médian | 391 | 536 | 83 | 93 | 59 | 67 | 44 | |
| FIND_GTR | 10 réels, cas médian | 390 | 535 | 83 | 93 | 59 | 67 | 44 | |
| FIND_LTR | 10 réels, cas médian | 390 | 535 | 83 | 93 | 59 | 67 | 44 | |
| MAX_ARR | 10 réels | 648 | 889 | 160 | 179 | 114 | 128 | 85 | |
| MIN_ARR | 10 réels | 601 | 825 | 148 | 164 | 105 | 118 | 78 | |
| OCCUR_ARR | 10 réels | 598 | 821 | 147 | 164 | 104 | 118 | 78 | |
| ROL_ARR | 10 réels | 273 | 374 | 67 | 75 | 48 | 54 | 35 | |
| ROR_ARR | 10 réels | 264 | 363 | 65 | 72 | 46 | 52 | 34 | |
| SORT_ARR | 10 réels | 896 | 1 229 | 220 | 245 | 156 | 176 | 116 | |
| EQUAL_ARR | 10 réels | 344 | 472 | 84 | 94 | 60 | 68 | 45 | |

Instructions de gestion du temps

Instructions de gestion des dates, heures et durées

Le tableau ci-après décrit les performances des instructions de gestion des dates, heures et durées.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-------------------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MW2:4 := ADD_DT(%MW2:4,%MD8) | 1 951 | 2 677 | 737 | 744 | 568 | 571 | 440 | 19 |
| %MD2 := ADD_TOD(%MD2,%MD8) | 1 025 | 1 407 | 387 | 391 | 298 | 300 | 231 | 9 |
| %MB2:11 := DATE_TO_STRING(%MD40) | 606 | 832 | 229 | 231 | 176 | 177 | 137 | 12 |
| %MW5 := DAY_OF_WEEK() | 88 | 121 | 33 | 34 | 26 | 26 | 20 | 5 |
| %MD10 := DELTA_D(%MD2, %MD4) | 731 | 1 004 | 276 | 279 | 213 | 214 | 165 | 9 |
| %MD10 := DELTA_DT(%MD2:4,%MW6:4) | 1 506 | 2 067 | 569 | 574 | 438 | 441 | 339 | 19 |
| %MD10 := DELTA_TOD(%MD2,%MD4) | 1 113 | 1 527 | 421 | 424 | 324 | 325 | 251 | 9 |
| %MB2:20 := DT_TO_STRING(%MW50:4) | 707 | 970 | 267 | 269 | 206 | 207 | 159 | 17 |
| %MW2:4 := SUB_DT(%MW2:4,%MD8) | 2 344 | 3 216 | 886 | 893 | 682 | 685 | 528 | 19 |
| %MD2 := SUB_TOD(%MD2,%MD8) | 1 113 | 1 527 | 421 | 424 | 324 | 325 | 251 | 9 |
| %MB2:15 := TIME_TO_STRING(%MD40) | 794 | 1 089 | 300 | 303 | 231 | 232 | 179 | 12 |
| %MB2:9 := TOD_TO_STRING(%MD40) | 519 | 712 | 196 | 198 | 151 | 152 | 117 | 12 |
| %MD100 := TRANS_TIME(%MD2) | 331 | 455 | 125 | 126 | 96 | 97 | 75 | 7 |

**Accès
horodateur**

Le tableau ci-après décrit les performances des instructions horodateur.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57• • |
| RRTC(%MW0:4) | 88 | 89 | 30 | 30 | 21 | 21 | 14 | 8 |
| WRTC(%MW0:4) | 69 | 70 | 25 | 25 | 17 | 17 | 11 | 8 |
| PTC(%MW0:5) | 74 | 75 | 26 | 27 | 18 | 19 | 12 | 8 |
| SCHEDULE(%MW0,%MW1,%MW2, %MD10,%MD12,%M0)ÿ | 88 | 89 | 30 | 30 | 21 | 21 | 14 | 8 |

**Fonctions
temporisateur**

Le tableau ci-après décrit les performances des fonctions temporisateur.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57• • |
| FTON | 53 | 53 | 28 | 32 | 21 | 24 | 12 | |
| FTOF | 53 | 53 | 28 | 32 | 21 | 24 | 12 | |
| FTP | 53 | 53 | 28 | 32 | 21 | 24 | 12 | |
| FPULSOR | 181 | 249 | 69 | 69 | 53 | 53 | 41 | |

Instructions sur chaînes de caractères

Affectation, copie de chaînes de caractères

Le tableau ci-après décrit les performances des instructions d'affectation et copie de chaînes de caractères

| ST | Taille (caractères) | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-------------------|---------------------|------------------------|------------|-----------|------------|-----------|------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MB0:8:=%MB10:8 | 8 caractères | 66 | 67 | 27 | 27 | 18 | 19 | 14 | 15 |
| | par caractère | 0,39 | 0,39 | 0,30 | 0,30 | 0,23 | 0,23 | 0,16 | |
| %MB0:8:='abcdefg' | 8 caractères | 85 | 85 | 29 | 29 | 20 | 20 | 14 | 14 |
| | par caractère | 2,37 | 2,37 | 0,68 | 0,68 | 0,47 | 0,47 | 0,36 | 0,5 |

Conversions mot <-> chaînes de caractères

Le tableau ci-après décrit les performances des instructions de conversion de mots et de chaînes de caractères.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---------------------------------|------------------------|------------|-----------|------------|-----------|------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MW1:= STRING_TO_INT(%MB0:7) | 69 | 71 | 23 | 23 | 16 | 16 | 12 | 10 |
| %MB0:7:= INT_TO_STRING(%MW0) | 74 | 75 | 23 | 23 | 15 | 16 | 12 | 10 |

**Conversions
double mot <->
chaînes de
caractères**

Le tableau ci-après décrit les performances des instructions conversion de double mots et de chaînes de caractères

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-----------------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MD1:= STRING_TO_DINT(%MB0:13) | 706 | 707 | 237 | 237 | 160 | 160 | 115 | 10 |
| %MB0:13:= DINT_TO_STRING(%MD0) | 215 | 216 | 66 | 67 | 44 | 45 | 33 | 10 |

**Conversions
flottant <->
chaînes de
caractères**

Le tableau ci-après décrit les performances des instructions conversion de flottant en de chaînes de caractères

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|-----------------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MF1:= STRING_TO_REAL(%MB0:15) | 1 912 | 1 913 | 344 | 344 | 237 | 237 | 155 | 10 |
| %MB0:15:= REAL_TO_STRING(%MF0) | 500 | 501 | 140 | 140 | 96 | 96 | 63 | 10 |

Instructions de manipulation de chaînes de caractères

Le tableau ci-après décrit les performances des instructions de manipulation de chaînes de caractères

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| %MB10:20 := CONCAT(%MB30:10,%MB50:10) | 494 | 678 | 187 | 188 | 144 | 144 | 111 | 24 |
| %MB10:20 := DELETE(%MB10:22,2,3); | 400 | 549 | 151 | 152 | 116 | 117 | 90 | 21 |
| %MW0 := EQUAL_STR(%MB10:20,%MB30:20); (le 5ème caractère diffère) | 338 | 463 | 128 | 129 | 98 | 99 | 76 | 19 |
| %MW0 := FIND(%MB10:20,%MB30:10); | 650 | 892 | 246 | 248 | 189 | 190 | 146 | 19 |
| %MB10:20 := INSERT(%MB30:10,%MB50:10,4); | 519 | 712 | 196 | 198 | 151 | 152 | 117 | 26 |
| %MB10:20 := LEFT(%MB30:30,20); | 369 | 506 | 139 | 141 | 107 | 108 | 83 | 19 |
| %MW0 := LEN(%MB10:20); | 219 | 300 | 83 | 83 | 64 | 64 | 49 | 12 |
| %MB10:20 := MID(%MB30:30,20,10); | 444 | 609 | 168 | 169 | 129 | 130 | 100 | 21 |
| %MB10:20 := REPLACE(%MB30:20,%MB50:10,10,1 0); | 556 | 763 | 210 | 212 | 162 | 163 | 125 | 28 |
| %MB10:20 := RIGHT(%MB30:30,20); | 606 | 832 | 229 | 231 | 176 | 177 | 137 | 19 |

Extraction de mots

Le tableau ci-après décrit les performances des instructions d'extraction de mots

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| LW | 32 | 44 | 12 | 12 | 9 | 9 | 7 | |
| HW | 32 | 44 | 12 | 12 | 9 | 9 | 7 | |
| CONCATW | 32 | 44 | 12 | 12 | 9 | 9 | 7 | |

Fonctions métier et fonctions Orphée

Fonctions de communication

Le tableau ci-après décrit les performances des fonctions de communication.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• |
| SEND_REQ : temps d'exécution de l'instruction, temps dans le système de communication à ajouter | | | | | | | | |
| SEND_REQ(%KW0:6,15,%MW0:1, %MW10:10,%MW30:4) | 1 250 | 1 715 | 472 | 476 | 364 | 366 | 282 | 33 |
| SEND_TLG : le traitement est synchrone de l'instruction, pas de temps dans le système de communication à ajouter | | | | | | | | |
| SEND_TLG(%KW0:6,,%MW0:5, %MW30:2) | 938 | 1 287 | 354 | 357 | 273 | 274 | 211 | 24 |
| SERVER pour 120 octets | 3 825 | 4 244 | 2 225 | 2 229 | 1 677 | 1 679 | 1 427 | 16 |
| WRITE_ASYN pour 500 mots | 2 975 | 3 301 | 1 731 | 1 734 | 1 305 | 1 306 | 1 110 | 16 |
| READ_ASYN pour 500 mots | 2 975 | 3 301 | 1 731 | 1 734 | 1 305 | 1 306 | 1 110 | 16 |

Fonction de dialogue opérateur

Le tableau ci-après décrit les performances des instructions dialogue opérateur.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• |
| SEND_MSG(ADR#1.0,%MW0:2,%MW10:2) | 1250 | 1715 | 472 | 476 | 364 | 366 | 282 | 25 |
| SEND_ALARM(ADR#1.0,%MW0:2,%MW10:2) | 1250 | 1715 | 472 | 476 | 364 | 366 | 282 | 25 |
| GET_MSG(ADR#1.0,%MW0:2,%MW10:2) | 1250 | 1715 | 472 | 476 | 364 | 366 | 282 | 25 |
| GET_VALUE(ADR#1.0,%MW0,%MW10:2) | 625 | 858 | 236 | 238 | 182 | 183 | 141 | 20 |
| ASK_MSG(ADR#1.0,%MW0:2,%MW10:2, %MW20:2) | 1250 | 1715 | 472 | 476 | 364 | 366 | 282 | 32 |
| ASK_VALUE(ADR#1.0,%MW0,%MW10:2, %MW20:2) | 1250 | 1715 | 472 | 476 | 364 | 366 | 282 | 27 |
| DISPLAY_ALRM(ADR#1.0,%MW0,%MW10:2) | 625 | 858 | 236 | 238 | 182 | 183 | 141 | 20 |
| DISPLAY_GRP(ADR#1.0,%MW0,%MW10:2) | 625 | 858 | 236 | 238 | 182 | 183 | 141 | 20 |
| DISPLAY_MSG(ADR#1.0,%MW0,%MW10:2) | 625 | 858 | 236 | 238 | 182 | 183 | 141 | 20 |

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--------------------------------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| CONTROL_LEDS(ADR#1.0,%MW0:2,%MW10:2) | 1250 | 1715 | 472 | 476 | 364 | 366 | 282 | 25 |
| ASSIGN_KEYS(ADR#1.0,%MW0:2,%MW10:2) | 1250 | 1715 | 472 | 476 | 364 | 366 | 282 | 25 |
| PANEL_CMD(ADR#1.0,%MW0:2,%MW10:2) | 1250 | 1715 | 472 | 476 | 364 | 366 | 282 | 25 |

Fonction de régulation

Le tableau ci-après décrit les performances des fonctions de régulation.

| ST | Condition | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---|-------------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| PID("PIDS1", 'Unité', %IW3.5,%MW12,%M16,%MW284:43) | deval_mmi=0 | 688 | 943 | 260 | 262 | 200 | 201 | 155 | 32 |
| | deval_mmi=1 | 563 | 772 | 213 | 214 | 164 | 165 | 127 | |
| PWM(%MW11,%Q2.1,%MW385:5) | | 313 | 429 | 118 | 119 | 91 | 91 | 70 | 17 |
| SERVO(%MW12,%IW3.6,%Q2.2,%Q2.3,%MW284:43,%MW390:10) | | 500 | 686 | 189 | 191 | 145 | 146 | 113 | 31 |
| PID_MMI(ADR#0.0.4,%M1,%M2:5,%MW410:62) | EN=1 | 625 | 858 | 236 | 238 | 182 | 183 | 141 | 30 |

Archivage de données

Le tableau ci-après décrit les performances des fonctions d'archivage de données.

| ST | Condition | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--------------|-----------|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | |
| SET_PCMCIA | fixe | | 350 | | 70 | | 40 | 30 | |
| | par mot | | 0,4 | | 0,3 | | 0,2 | 0,2 | |
| WRITE_PCMCIA | fixe | | 350 | | 70 | | 40 | 30 | |
| | par mot | | 0,8 | | 0,3 | | 0,3 | 0,2 | |
| READ_PCMCIA | fixe | | 350 | | 70 | | 40 | 30 | |
| | par mot | | 0,7 | | 0,4 | | 0,3 | 0,4 | |

Fonction Orphée Le tableau ci-après décrit les performances des fonctions de régulation.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|--|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• |
| DSHL_RBIT(%MD102,16,%MD204,%MD206) | 200 | 274 | 76 | 76 | 58 | 58 | 45 | 17 |
| DSHR_RBIT(%MD102,16,%MD204,%MD206) | 319 | 437 | 120 | 122 | 93 | 93 | 72 | 17 |
| DSHRZ_C(%MD102,16,%MD204,%MD206) | 194 | 266 | 73 | 74 | 56 | 57 | 44 | 17 |
| WSHL_RBIT(%MW102,8,%MW204,%MW206) | 138 | 189 | 52 | 52 | 40 | 40 | 31 | 17 |
| WSHR_RBIT(%MW102,8,%MW204,%MW206) | 181 | 249 | 69 | 69 | 53 | 53 | 41 | 17 |
| WSHRZ_C(%MW102,8,%MW204,%MW206) | 138 | 189 | 52 | 52 | 40 | 40 | 31 | 17 |
| SCOUNT(%M100,%MW100,%M101,%M102,%MW101,%MW102,%M200,%M201,%MW200,%MW201) | 263 | 360 | 99 | 100 | 76 | 77 | 59 | 38 |

Instructions d'entrées/sorties explicite

Performances Le tableau ci-après décrit les performances des instructions d'entrées/sorties explicites.

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• |
| Read_Sts %CHi.MOD | | | | | | | | |
| Quelque soit le métier excepté la voie de communication du processeur | 552 | 651 | 291 | 292 | 220 | 220 | | 2 |
| Read_Sts %CHi | | | | | | | | |
| EntréeTOR | 296 | 317 | 180 | 181 | 136 | 136 | 117 | 6 |
| Sortie TOR | 386 | 426 | 227 | 227 | 171 | 171 | 145 | |
| Entrée analogique | 334 | 363 | 201 | 201 | 151 | 151 | 129 | |
| Sortie analogique | 327 | 354 | 197 | 197 | 148 | 148 | 127 | |
| Module de comptage CTY 2A/4A | 327 | 354 | 197 | 197 | 148 | 148 | 127 | |
| Module de comptage CTY 2C | 311 | 335 | 189 | 189 | 142 | 142 | 122 | |
| Module pas à pas CFY 11/22 | 448 | 505 | 254 | 255 | 192 | 192 | 163 | |
| Module commande d'axe CAY 21/41 | 334 | 363 | 201 | 201 | 151 | 151 | 129 | |
| Write_Param %CHi | | | | | | | | |
| Entrée analogique | 499 | 574 | 274 | 275 | 207 | 207 | 174 | 6 |
| Sortie analogique | 474 | 540 | 265 | 265 | 200 | 200 | 169 | |
| Module de comptage CTY 2A/4A | 603 | 731 | 302 | 303 | 229 | 230 | 190 | |
| Module de comptage CTY 2C | 400 | 444 | 233 | 234 | 176 | 176 | 150 | |
| Module pas à pas CFY 11/22 | 577 | 691 | 297 | 298 | 225 | 226 | 187 | |
| Module commande d'axe CAY 21/41 | 461 | 522 | 260 | 260 | 196 | 196 | 166 | |
| Read_Param %CHi | | | | | | | | |
| Entrée analogique | 115 | 118 | 75 | 75 | 56 | 56 | 49 | 6 |
| Sortie analogique | 115 | 118 | 75 | 75 | 56 | 56 | 49 | |
| Module de comptage CTY 2A/4A | 334 | 363 | 201 | 201 | 151 | 151 | 129 | |
| Module de comptage CTY 2C | 349 | 381 | 209 | 209 | 157 | 157 | 134 | |
| Module pas à pas CFY 11/22 | 393 | 435 | 230 | 230 | 173 | 173 | 148 | |
| Module commande d'axe CAY 21/41 | 386 | 426 | 227 | 227 | 171 | 171 | 145 | |
| Save_Param %CHi | | | | | | | | |

| ST | Temps d'exécution (µs) | | | | | | | Taille (mots) |
|---|------------------------|---------------|--------------|---------------|--------------|---------------|-------|---------------|
| | 57 1• ram | 57 1• cart | 57 2• ram | 57 2• cart | 57 3• ram | 57 3• cart | 57 4• | 57•• |
| Entrée analogique | 635 | 787 | 306 | 307 | 232 | 233 | 191 | 6 |
| Sortie analogique | 640 | 795 | 306 | 307 | 233 | 233 | 191 | |
| Module de comptage CTY 2A/4A | 752 | 1 049 | 271 | 274 | 209 | 211 | 160 | |
| Module de comptage CTY 2C | 379 | 417 | 223 | 223 | 168 | 168 | 143 | |
| Module pas à pas CFY 11/22 | 421 | 470 | 243 | 243 | 183 | 183 | 155 | |
| Module commande d'axe CAY 21/41 | 421 | 470 | 243 | 243 | 183 | 183 | 155 | |
| Restore_Param %CHi | | | | | | | | |
| Entrée analogique | 467 | 531 | 262 | 263 | 198 | 198 | 167 | 6 |
| Sortie analogique | 467 | 531 | 262 | 263 | 198 | 198 | 167 | |
| Module de comptage CTY 2A/4A | 608 | 739 | 303 | 304 | 230 | 230 | 190 | |
| Module de comptage CTY 2C | 349 | 381 | 209 | 209 | 157 | 157 | 134 | |
| Module pas à pas CFY 11/22 | 588 | 707 | 299 | 300 | 227 | 227 | 188 | |
| Module commande d'axe CAY 21/41 | 480 | 548 | 267 | 268 | 202 | 202 | 170 | |
| Write_Cmd %CHi | | | | | | | | |
| Sortie TOR | 288 | 308 | 176 | 176 | 133 | 133 | 114 | 6 |
| Entrées analogique | | | | | | | 0 | |
| Forçage entrée | 134 | 138 | 86 | 86 | 65 | 65 | 56 | |
| Recalibration entrée | 691 | 895 | 303 | 304 | 231 | 232 | 186 | |
| Sortie analogiques | | | | | | | 0 | |
| Forçage | 143 | 147 | 92 | 92 | 69 | 69 | 60 | |
| Smove %CH1.0(%MW1,%MW2,%MW3,%MD4,%MD5,%MW6) | | | | | | | | |
| Module pas à pas CFY 11/22 | 617 | 755 | 304 | 305 | 231 | 231 | 190 | 19 |
| Module commande d'axe CAY 21/41 | 608 | 739 | 303 | 304 | 230 | 230 | 190 | |

Bloc fonction DFB

Taille occupée par le type DFB

La formule suivante permet de calculer la taille occupée par le type DFB :

Taille type DFB = Taille variables et paramètres DFB + Taille code DFB

Taille variables et paramètres DFB

Le calcul de la taille des variables et des paramètres DFB s'effectue de la façon suivante :

Taille variables et paramètres DFB = 110 + Somme des descripteurs des variables et paramètres + Somme des tailles occupées par chaque variable ou paramètre

avec :

Descripteur d'une variable ou d'un paramètre = $5,5 + (\text{Nombre de caractères du nom de la variable ou du paramètre})/2$

et

Taille occupée par chaque variable ou paramètre :

| Type | IN | IN/OUT | OUT | PUBLIC | PRIVEE |
|--------|-----|--------|-------|--------|--------|
| EBOOL | 0,5 | 2 | 0,5 | 0,5 | 0,5 |
| BOOL | 0,5 | 2 | 0,5 | 0,5 | 0,5 |
| WORD | 1 | 2 | 1 | 1 | 1 |
| DWORD | 2 | 2 | 2 | 2 | 2 |
| REAL | 2 | 2 | 2 | 2 | 2 |
| AR_X | 3 | 3 | 0,5*N | 0,5*N | 0,5*N |
| AR_W | 3 | 3 | N | N | N |
| AR_D | 3 | 3 | 2*N | 2*N | 2*N |
| AR_R | 3 | 3 | 2*N | 2*N | 2*N |
| STRING | 3 | 3 | 0,5*N | 0,5*N | 0,5*N |

N = nombre d'éléments du tableau ou longueur de la chaîne de caractères (STRING)

Taille code DFB Le calcul de la taille des variables et des paramètres DFB s'effectue de la façon suivante :

Taille code DFB = 11 + Somme des tailles de chacune des instructions (1)

(1) Au taille des instructions rajouter les volumes suivant en fonction de la variable ou du paramètre contenu dans l'instruction :

| Nature | Type | Volume |
|--------------------------|----------------------------|--------|
| IN | EBOOL | 0,5 |
| | BOOL, WORD, DWORD, REAL | 0 |
| | AR_X,AR_W,AR_D,AR_R,STRING | 3 |
| IN/OUT | EBOOL | 3 |
| | BOOL, WORD, DWORD, REAL | 3 |
| | AR_X,AR_W,AR_D,AR_R,STRING | 3 |
| OUT, PUBLIC PRIVEE | EBOOL | 0,5 |
| | BOOL, WORD, DWORD, REAL | 0 |
| | AR_X,AR_W,AR_D,AR_R,STRING | 0 |

Objet indexé:

| Nature | Type | Volume |
|---------------------------|---------------------|--------|
| IN | AR_X,AR_W,AR_D,AR_R | 7 |
| IN/OUT | AR_X,AR_W,AR_D,AR_R | 7 |
| OUT, PUBLIC, PRIVEE | AR_X,AR_W,AR_D,AR_R | 6 |

Taille occupée par l'utilisation du DFB

Appel d'une instance de DFB sans aucun paramètre = 6 mots

Calcul pour un paramètre

| Nature | Type | Volume |
|--------|--------------------------------|---------------------|
| IN | EBOOL, BOOL, WORD, DWORD, REAL | idem affectation := |
| | AR_X,AR_W,AR_D,AR_R, STRING | 14 |
| IN/OUT | EBOOL, WORD, DWORD, REAL | 10 |
| | BOOL, AR_X,AR_W,AR_D,AR_R | 14 |
| OUT, | Tout type | idem affectation := |

Utilisation d'une variable d'une instance : rajouter 1 mot par rapport

Temps d'exécution

Le temps total d'exécution du DFB est donné par la formule suivante :

Temps total d'exécution du DFB = Talon code DFB + Somme des temps d'accès aux variables et paramètres DFB + Appel DFB (sans paramètre) + Somme des temps d'accès de chaque paramètre

Le tableau suivant fournit les temps d'exécution en μ s.

| Élément | Type | 571• ram | 571• cart | 572• ram | 572• cart | 573• ram | 573• cart | 574• |
|---|---------------------------------|-------------|--------------|-------------|--------------|-------------|--------------|------|
| Talon pour code DFB | | 13,0 | 16,3 | 4,8 | 4,8 | 3,4 | 3,8 | 1,1 |
| Accès à une variable ou un paramètre DFB (1) | | | | | | | | |
| Objets indexés | | | | | | | | |
| IN | EBOOL | 0,2 | 0,3 | 0,09 | 0,12 | 0,06 | 0,10 | 0,02 |
| | BOOL,WORD,DWORD,REAL | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | AR_X,AR_W,AR_D,AR_R,STRING | 1,4 | 1,7 | 0,5 | 0,8 | 0,4 | 0,6 | 0,1 |
| IN/OUT | | 1,4 | 1,7 | 0,5 | 0,8 | 0,4 | 0,6 | 0,1 |
| OUT, PUBLIC, PRIVEE | EBOOL | 0,2 | 0,3 | 0,09 | 0,12 | 0,06 | 0,10 | 0,02 |
| | BOOL,WORD,DWORD,REAL | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | AR_X,AR_W,AR_D,AR_R,STRING | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Objets non indexés | | | | | | | | |
| IN, IN/ OUT | AR_X,AR_W,AR_D,AR_R, | 2,8 | 3,5 | 1,0 | 0,9 | 0,7 | 0,7 | 0,2 |
| OUT, PUBLIC, PRIVEE | AR_X,AR_W,AR_D,AR_R, | 2,1 | 2,6 | 0,8 | 1,1 | 0,5 | 0,9 | 0,2 |
| Appel DFB (sans paramètre) | | 3,8 | 4,8 | 1,5 | 1,8 | 1,0 | 1,4 | 0,3 |
| Calcul par paramètre (1) | | | | | | | | |
| IN | EBOOL | idem := | | | | | | |
| | BOOL,WORD,DWORD,REAL | idem := | | | | | | |
| | AR_X,AR_W,AR_D,AR_R,STRING | 3,5 | 4,4 | 1,5 | 1,9 | 1,0 | 1,6 | 0,3 |
| IN/OUT | EBOOL,WORD,DWORD,REAL | 2,8 | 3,5 | 1,0 | 1,4 | 0,7 | 1,1 | 0,2 |
| | BOOL,AR_X,AR_W,AR_D,AR_R,STRING | 3,5 | 4,4 | 1,5 | 1,9 | 1,0 | 1,6 | 0,3 |
| OUT | Tout type | idem := | | | | | | |

(1) valeur à ajouter par rapport aux opérations s'appliquant à des objets de type %M.

6.4 Fonctions avancées

Présentation

Objet de ce sous chapitre Ce sous chapitre décrit les tailles mémoire occupées par les fonctions avancées et la méthode de calcul du nombre d'instructions.

Contenu de ce sous-chapitre Ce sous-chapitre contient les sujets suivants :

| Sujet | Page |
|---|------|
| Description des zones mémoire | 199 |
| Taille mémoire des objets | 200 |
| Bilan des consommations mémoire des modules sur Micro | 201 |
| Bilan des consommations mémoire des modules sur Premium | 204 |
| Taille mémoire des fonctions avancées | 210 |
| Méthode de calcul du nombre d'instructions | 220 |

Description des zones mémoire

Rappel

L'application se répartit en plusieurs zones mémoire:

- zone mémoire bit :
 - cette zone est spécifique pour les automates TSX 37 et est limité à 1280 bits
 - cette zone fait partie de la zone mémoire donnée pour les automates TSX 57,
- zone mémoire données (mots),
- zone mémoire application, comprenant :
 - la configuration,
 - le programme,
 - les constantes.

Les zones mémoire bits et données sont toujours stockées en RAM interne, la zone mémoire application peut être stockée en RAM interne ou sur carte mémoire.

Taille mémoire des objets

Description

La table suivante décrit la taille mémoire occupée par chaque type d'objet du langage PL7.

| Type d'objet | Mémoire bit (en mots) | Données (en mots) | Application (en mots) |
|--------------------------------|-----------------------|-------------------|-----------------------|
| Etapes Grafcet (%Xi, %Xi.T) | 0,5 | 1 | |
| %Mi | 0,5 | | |
| Numériques (%MWi) | | 1 | |
| Constantes (%KW _i) | | | 1,25 |
| %NWi | | 1 | |
| %Ti | | 4 | 2 |
| %TMi | | 5 | 2 |
| %MNi | | 4 | 2 |
| %Ci | | 3 | 1 |
| %Ri (longueur lg) | | 6+lg | 2 |
| %DRi | | 6 | 49 |

Données de l'interpréteur Grafcet = $355 + 2 \times \text{Nb étapes actives configurées} + (\text{Nb de transitions valides configurées}) / 2$

Bilan des consommations mémoire des modules sur Micro

Généralités

Note : Ces informations sont données pour une version particulière de processeurs. Elles peuvent subir de «légères» variations en fonction des évolutions du produit.

Pour chaque type de modules, les tableaux suivants fournissent la taille occupée dans chacune des zones ainsi qu'une taille fixe à ajouter au bilan de consommation la première fois qu'une fonction métier est utilisée.

Processeurs

Le tableau ci-après fournit les consommations mémoire des modules processeurs TSX 37.

| Processeurs | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|---------------------------------------|---------------------|----------------|-------------------------|
| TSX 37-05/08/10 | 70 | 1560 | 920 |
| TSX 37-21 | 70 | 1570 | 930 |
| TSX 37-22 | 70 | 2110 | 1280 |
| Utilisation tâche FAST (TSX 37) | | 260 | |
| Utilisation premier événement(TSX 37) | | 520 | |

Modules Tout ou rien

Le tableau ci-après fournit les consommations mémoire des modules Tout ou Rien.

| Famille Tout ou Rien | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|----------------------|---------------------|----------------|-------------------------|
| 8 entrées TOR | 4 | 12 | 40 |
| 16 entrées TOR | 8 | 12 | 50 |
| 4 sorties TOR | 2 | 12 | 40 |
| 8 sorties TOR | 4 | 12 | 40 |
| 8E / 8S TOR | 4 | 12 | 40 |
| 16E / 12S TOR | 16 | 20 | 100 |
| 32E / 32S TOR | 32 | 20 | 142 |

Modules analogiques

Le tableau ci-après fournit les consommations mémoire des modules analogiques.

| Famille analogique | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|--|---------------------|----------------|-------------------------|
| Module 4 entrées analogiques | | | |
| AEZ414 | 0 | 156 | 56 |
| Surcoût 1er module Famille 4 Entrées analogiques | | | 120 |
| Module 8 entrées analogiques | | | |
| AEZ801/AEZ802 | 0 | 212 | 72 |
| Surcoût 1er module Famille 8 Entrées analogiques | | | 120 |
| Module sorties analogiques | | | |
| ASZ200 | 0 | 52 | 40 |
| ASZ401 | 0 | 100 | 59 |
| Surcoût 1er module Famille Sorties analogiques | | | 120 |

Modules de comptage

Le tableau ci-après fournit les consommations mémoire des modules de comptage.

| Famille Comptage | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|---------------------------------|---------------------|----------------|-------------------------|
| CTY1A | 16 | 108 | 64 |
| CTY2A | 32 | 212 | 106 |
| Surcoût 1ère voie en Comptage | | | 144 |
| Surcoût 1ère voie en Décomptage | | | 144 |
| Surcoût 1ère voie en CPT/DCPT | | | 144 |

Modules de communication

Le tableau ci-après fournit les consommations mémoire des modules de communication.

| Famille Communication | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|--------------------------------------|----------------------------|-----------------------|--------------------------------|
| STZ010 | 0 | 36 | 168 |
| SCP111/SCP112/SCP114 (sur UC UTW) | 0 | 40 | 763 |
| FPP 20 sur UC (Voie 0 UTW) | 0 | 40 | 755 |
| MDM 10 | 0 | 2528 | 12880 |

Bilan des consommations mémoire des modules sur Premium

Généralités

Note : Ces informations sont données pour une version particulière de processeurs. Elles peuvent subir de « légères » variations en fonction des évolutions du produit.

Pour chaque type de modules, les tableaux suivants fournissent la taille occupée dans chacune des zones ainsi qu'une taille fixe à ajouter au bilan de consommation la première fois qu'une fonction métier est utilisée.

Processeurs

Le tableau ci-après fournit les consommations mémoire des modules processeurs TSX 57.

| Processeurs | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|--|---------------------|----------------|-------------------------|
| P 57-1• | 70 | 4714 | 1720 |
| P 57-2•/3•/4• | 70 | 4714 | 1784 |
| Utilisation tâche FAST (TSX 57) | | 520 | |
| Surcoût 1er module dans configuration | | 600 | |
| P 57-1• : par boucle de régulation | | 500 | |
| Surcoût 1er boucle | | | 25000 |
| P 57-2•/3•/4• : par boucle de régulation | | 500 | |
| Surcoût 1er boucle | | | 5000 |

Modules TOR

Le tableau ci-après fournit les consommations mémoire des modules Tout ou Rien..

| Famille TOR | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|--|----------------------------|-----------------------|--------------------------------|
| Famille Entrées TOR simples | | | |
| 8 entrées TOR | 4 | 100 | 100 |
| 16 entrées TOR | 8 | 130 | 110 |
| 32 Entrées TOR | 16 | 230 | 120 |
| 64 Entrées TOR | 32 | 430 | 190 |
| Surcoût 1er module Famille Entrées | | | 610 |
| Famille Sorties TOR simples | | | |
| 8 Sorties TOR | 4 | 110 | 100 |
| 16 Sorties TOR | 8 | 160 | 110 |
| 32 Sorties TOR | 16 | 280 | 120 |
| 64 Sorties TOR | 32 | 550 | 190 |
| Surcoût 1er module Famille Sorties | | | 570 |
| Famille Entrées TOR événementielles | | | |
| 16 Entrées TOR (DEY 16FK) | 8 | 220 | 130 |
| Surcoût 1er module Famille Entrées | | | 680 |
| Famille Entrées/Sorties TOR sécurité | | | |
| 12E/4S ou 12E/2S(PAY) | 16 | 128 | 200 |
| Surcoût 1er module Famille Entrées TOR EVT | | | 1320 |
| Famille entrées/sorties TOR mixte | | | |
| 16 Entrées/12 sorties (DMY 28FK) | 16 | 304 | 152 |
| Surcoût 1er module Famille | | | 1432 |
| TOR mixte | | | |
| 16 E/12 S réflexe (DMY 28RFK) | 32 | 976 | 656 |
| Surcoût 1er module Famille TOR mixte réflexe | | | 5596 |

Modules analogiques

Le tableau ci-après fournit les consommations mémoire des modules analogiques.

| Famille analogique | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|---|---------------------|----------------|-------------------------|
| Familles Entrées Analogiques | | | |
| AEY414 | 4 | 430 | 160 |
| AEY800 | 8 | 840 | 240 |
| AEY1600 | 16 | 1670 | 430 |
| Surcoût 1er module Famille Entrées Analogiques (AEY 414/800/1600) | | | 2990 |
| AEY810 | 8 | 888 | 248 |
| AEY1614 | 16 | 1768 | 432 |
| Surcoût 1er module Famille Entrées Analogiques (AEY 810/1614) | | | 3056 |
| AEY420 | 4 | 476 | 168 |
| Surcoût 1er module Famille Entrées Analogiques (AEY 810/1614) | | | 2080 |
| Famille Sorties Analogiques | | | |
| ASY410 | 4 | 430 | 160 |
| Surcoût 1er module Sorties Analogiques ASY410 | | | 1700 |
| ASY800 | 8 | 744 | 248 |
| Surcoût 1er module Sorties Analogiques ASY800 | | | 1760 |

Modules comptage

Le tableau ci-après fournit les consommations mémoire des modules de comptage.

| Famille Comptage | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|---------------------------------------|---------------------|----------------|-------------------------|
| Module CTY2A | 32 | 410 | 170 |
| Module CTY4A | 64 | 800 | 250 |
| Surcoût 1ère voie comptage configurée | | | 1740 |
| Module CTY2C | 48 | 672 | 184 |
| Surcoût 1ère voie comptage configurée | | | 1992 |

Modules servo-moteur

Le tableau ci-après fournit les consommations mémoire des modules servo-moteur.

| Famille Servo-Moteur | Mémoire bits (mots) | (mots) | (mots) |
|---------------------------------------|---------------------|--------|--------|
| CAY•1 | 78 | 520 | 140 |
| CAY•2 | 78 | 376 | 232 |
| CAY33 voie 3 | 78 | 264 | 170 |
| Surcoût 1ère voie configurée CAY•1 | | | 2130 |
| Surcoût 1ère voie configurée CAY•2/33 | | | 3600 |
| Surcoût 1ère voie 3 configurées CAY33 | | | 3600 |

Modules pas à pas

Le tableau ci-après fournit les consommations mémoire des modules de pas à pas.

| Famille pas à pas | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|--|---------------------|----------------|-------------------------|
| CFY11 | 29 | 323 | 104 |
| CFY21 | 58 | 646 | 152 |
| Surcoût 1ère voie pas à pas configurée | | | 2368 |

Modules de communication

Le tableau ci-après fournit les consommations mémoire des modules de communication.

| Famille Module de communication | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|--------------------------------------|---------------------|----------------|-------------------------|
| SCY21600 (Voie 0 UTW) | 1 | 230 | 80 |
| sur SCY21600 (Voie 1 UTW) | 1 | 450 | 40 |
| Surcoût 1ère voie configurée en UTW | | | 1280 |
| ETY 110 | 1 | 431 | 256 |
| Surcoût 1ère voie configurée ETY 110 | | | 1984 |
| ETY 120 | 1 | 48 | 136 |
| Surcoût 1ère voie configurée ETY 120 | | | 1368 |
| ETY 210 | 1 | 434 | 400 |
| Surcoût 1ère voie configurée ETY 210 | | | 3424 |
| IBY 100 | 1 | 450 | 40 |

Sous modules de communication

Le tableau ci-après fournit les consommations mémoire des sous-modules de communication.

| Sous module de communication | Mémoire bits (mots) | Données (mots) | Zone application (mots) |
|---|---------------------|----------------|-------------------------|
| SCP111/ SCP112/ SCP114 (UTW) sur UC (Voie 0 UTW) | 1 | 60 | 580 |
| FPP 20 sur UC (Voie 0 UTW) | 1 | 60 | 580 |
| FPP 10 sur UC (Voie 0 UTW) | 1 | 40 | 870 |

Coupleur AS-i

Le tableau ci-après fournit les consommations mémoire des coupleur AS-i.

| Famille AS-i | Mémoire bits (mots) | Données (mots) | Zone application (mots) |
|-----------------------|---------------------|----------------|-------------------------|
| SAY | 3 | 373 | 176 |
| Surcoût 1ère voie ASi | | | 2272 |

Modules de pesage

Le tableau ci-après fournit les consommations mémoire des modules de pesage.

| Famille Pesage | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|-------------------------------------|---------------------|----------------|-------------------------|
| AWY001 | 1 | 170 | 120 |
| Surcoût 1ère voie pesage configurée | | | 3920 |

**Modules
d'entrées/sorties
déportées TBX**

Le tableau ci-après fournit les consommations mémoire des modules d'entrées/sorties déportées TBX

| Famille entrées/sorties déportés | Mémoire bits (mots) | Données (mots) | Zone Application (mots) |
|----------------------------------|---------------------|----------------|-------------------------|
| Entrées TOR | 8 | 152 | 88 |
| Surcoût 1ère embase configurée | | | 1400 |
| Sorties TOR | 8 | 176 | 88 |
| Surcoût 1ère embase configurée | | | 1320 |
| Programmable | 8 | 160 | 88 |
| Surcoût 1ère embase configurée | | | 2304 |
| Mémorisation d'état | 8 | 160 | 88 |
| Surcoût 1ère embase configurée | | | 1400 |
| AES 400 | 2 | 270 | 104 |
| ASS 200 | 2 | 270 | 104 |
| AMS 620 | 4 | 508 | 112 |
| Surcoût 1ère embase configurée | | | 3968 |

**Modules
Momentum**

Le tableau ci-après fournit les consommations mémoire des modules Momentum

| Famille Momentum | Mémoire bits (mots) | Données (mots) | Zone application (mots) |
|--------------------------------|---------------------|----------------|-------------------------|
| Entrées | 16 | 96 | 72 |
| Surcoût 1ère embase configurée | | | 1384 |
| Sortie | 16 | 112 | 72 |
| Surcoût 1ère embase configurée | | | 1256 |
| Mixte | 16 | 104 | 72 |
| Surcoût 1ère embase configurée | | | 1424 |

**Modules déport
bus X**

Le tableau ci-après fournit les consommations mémoire du module de déport bus X.

| Déport bus X | Mémoire bits (mots) | Données (mots) | Zone application (mots) |
|--------------------|---------------------|----------------|-------------------------|
| Module TSX REY 200 | 0 | 0 | 56 |

Taille mémoire des fonctions avancées

Description

Les tableaux suivants indiquent pour chaque fonction avancée (OF), la taille du code embarqué dans l'application (zone application) sur appel d'une fonction avancée.

Les fonctions d'une même famille partagent du code (code commun). Ce code commun est embarqué dans l'automate au premier appel d'une fonction de cette famille. Le code spécifique à une fonction est embarqué au premier appel de cette fonction.

Exemple

- Premier appel d'une fonction de la famille Conversions numériques, cas de DBCD_TO_DINT, code embarqué dans la zone application :
 - Code commun = 154 mots
 - Code OF DBCD_TO_INT = 149 mots
- Appel d'une autre fonction de la famille Conversions numériques, cas de DINT_TO_DBCD, code embarqué dans la zone application :
 - Code OF DINT_TO_DBCD = 203 mots
 - Appel d'une fonction de la famille Conversions numériques déjà appelée (DBCD_TO_DINT ou DINT_TO_DBCD) : aucun code embarqué

Conversions numériques

Le tableau suivant donne la consommation mémoire des fonctions avancées de conversion.

| Conversions numériques | OF | Taille code (en mots) |
|--|--------------|-----------------------|
| Conversion d'un nombre BCD 32 bits en entier 32 bits | DBCD_TO_DINT | 203 |
| Conversion d'un nombre BCD 32 bits en entier 16 bits | DBCD_TO_INT | 149 |
| Conversion d'un entier 32 bits en nombre BCD 32 bits | DINT_TO_DBCD | 203 |
| Conversion d'un entier 16 bits en nombre BCD 32 bits | INT_TO_DBCD | 75 |
| Extraction du mot de poids faible d'un double mot | LW | 33 |
| Extraction du mot de poids fort d'un double mot | HW | 33 |
| Formation d'un double mot avec 2 mots | CONCATW | 33 |
| | code commun | 154 |

Instructions sur chaînes de bits

Le tableau suivant donne la consommation mémoire des fonctions avancées de chaîne de bits.

| Chaînes de bits | OF | Taille code |
|--|-------------|-------------|
| ET logique entre deux tableaux | AND_ARX | 209 |
| Copie d'un tableau de bits dans un tableau de mots doubles | BIT_D | 248 |
| Copie d'un tableau de bits dans un tableau de mots | BIT_W | 205 |
| Copie d'un tableau de bits dans un tableau de bits | COPY_BIT | 146 |
| Copie d'un tableau de mots doubles dans un tableau de bits | D_BIT | 196 |
| Complément à un d'un tableau | NOT_ARX | 157 |
| OU logique entre deux tableaux | OR_ARX | 209 |
| Copie d'un tableau de mots dans un tableau de bits | W_BIT | 195 |
| OU exclusif entre deux tableaux | XOR_ARX | 209 |
| Longueur en nombre d'éléments | LENGTH_ARX | 20 |
| | code commun | 427 |

Instructions sur Tableau de mots

Le tableau suivant donne la consommation mémoire des fonctions avancées sur tableaux de mots.

| Instructions sur tableaux de mots | OF | Taille code (en mots) |
|--|-------------|-----------------------|
| Recherche du 1er élément d'un tableau égal à une valeur | FIND_EQW | 75 |
| Recherche du 1er élément d'un tableau supérieur à une valeur | FIND_GTW | 75 |
| Recherche du 1er élément d'un tableau inférieur à une valeur | FIND_LTW | 78 |
| Recherche de la plus grande valeur dans un tableau | MAX_ARW | 78 |
| Recherche de la plus petite valeur dans un tableau | MIN_ARW | 74 |
| Nombre d'occurrences d'une valeur dans un tableau | OCCUR_ARW | 145 |
| Décalage circulaire à gauche d'un tableau | ROL_ARW | 150 |
| Décalage circulaire à droite d'un tableau | ROR_ARW | 144 |
| Tri d'un tableau (croissant ou décroissant) | SORT_ARW | 164 |
| Recherche partielle du 1 ^o élément d'un tableau égal à une valeur | FIND_EQWP | 77 |
| Longueur en nombre d'éléments | LENGTH_ARW | 20 |
| | code commun | 162 |

Instructions sur tableaux de doubles-mots

Le tableau suivant donne la consommation mémoire des fonctions avancées sur tableaux de doubles mots

| Instructions sur tableaux de doubles-mots | OF | Taille code (en mots) |
|--|-------------|------------------------------|
| Recherche du 1er élément d'un tableau égal à une valeur | FIND_EQD | 79 |
| Recherche du 1er élément d'un tableau supérieur à une valeur | FIND_GTD | 80 |
| Recherche du 1er élément d'un tableau inférieur à une valeur | FIND_LTD | 95 |
| Recherche de la plus grande valeur dans un tableau | MAX_ARD | 95 |
| Recherche de la plus petite valeur dans un tableau | MIN_ARD | 78 |
| Nombre d'occurrences d'une valeur dans un tableau | OCCUR_ARD | 163 |
| Décalage circulaire à gauche d'un tableau | ROL_ARD | 170 |
| Décalage circulaire à droite d'un tableau | ROR_ARD | 178 |
| Tri d'un tableau (croissant ou décroissant) | SORT_ARD | |
| Recherche partielle du 1 ^o élément d'un tableau égal à une valeur | FIND_EQWP | 77 |
| Longueur en nombre d'éléments | LENGTH_ARW | 20 |
| | code commun | 162 |

Instructions sur tableaux de flottants

Le tableau suivant donne la consommation mémoire des fonctions avancées sur tableau de flottants

| Instructions sur tableaux de flottants | OF | Taille code (en mots) |
|--|-----------|------------------------------|
| Somme des éléments d'un tableau de réels | SUM_ARR | 152 |
| Recherche du 1er élément d'un tableau égal à une valeur | FIND_EQR | 134 |
| Recherche du 1er élément d'un tableau égal à une valeur à partir d'un rang | FIND_EQRP | 135 |
| Recherche du 1er élément d'un tableau supérieur à une valeur | FIND_GTR | 134 |
| Recherche du 1er élément d'un tableau inférieur à une valeur | FIND_LTR | 134 |
| Recherche de la plus grande valeur dans un tableau | MAX_ARR | 161 |
| Recherche de la plus petite valeur dans un tableau | MIN_ARR | 162 |
| Nombre d'occurrences d'une valeur dans un tableau | OCCUR_ARR | 132 |

| Instructions sur tableaux de flottants | OF | Taille code (en mots) |
|---|-------------|------------------------------|
| Décalage circulaire à gauche d'un tableau | ROL_ARR | 167 |
| Décalage circulaire à droite d'un tableau | ROR_ARR | 173 |
| Tri d'un tableau (croissant ou décroissant) | SORT_ARR | 271 |
| comparaison de 2 tableaux de réel | EQUAL_ARR | 173 |
| Tri d'un tableau (croissant ou décroissant) | LENGTH_ARR | 20 |
| | code commun | 124 |

Instructions de gestion du temps

Le tableau suivant donne la consommation mémoire des fonctions avancées de gestion du temps.

| Dates, heures et durées | OF | Taille code |
|--|----------------|--------------------|
| Ajout d'une durée à une date complète | ADD_DT | 519 |
| Ajout d'une durée à une heure du jour | ADD_TOD | 188 |
| Conversion d'une date en chaîne | DATE_TO_STRING | 150 |
| Jour de la semaine | DAY_OF_WEEK | 99 |
| Ecart entre deux dates | DELTA_D | 374 |
| Ecart entre deux dates complètes | DELTA_DT | 547 |
| Ecart entre deux heures du jour | DELTA_TOD | 110 |
| Conversion d'une date complète en chaîne | DT_TO_STRING | 266 |
| Retrait d'une durée à une date complète | SUB_DT | 548 |
| Retrait d'une durée à une heure du jour | SUB_TOD | 186 |
| Conversion d'une durée en chaîne | TIME_TO_STRING | 413 |
| Conversion d'une heure du jour en chaîne | TOD_TO_STRING | 156 |
| Mise sous la forme heures-mn-sec d'une durée | TRANS_TIME | 211 |
| Fonction horodateur | SCHEDULE | 700 |
| | code commun | 1703 |

Instructions sur chaîne de caractères

Le tableau suivant donne la consommation mémoire des fonctions avancées chaîne de caractères.

| Dates, heures et durées | OF | Taille code |
|---|-------------|-------------|
| Instructions sur chaînes de caractères | | taille code |
| Concaténation de deux chaînes | CONCAT | |
| Suppression d'une sous-chaîne | DELETE | 279 |
| Recherche du premier caractère différent | EQUAL_STR | 212 |
| Recherche d'une sous-chaîne | FIND | 225 |
| Insertion d'une sous-chaîne | INSERT | 287 |
| Extraction de la partie gauche d'une chaîne | LEFT | 38 |
| Longueur d'une chaîne | LEN | 70 |
| Extraction d'une sous-chaîne | MID | 44 |
| Remplacement d'une sous-chaîne | REPLACE | 365 |
| Extraction de la partie droite d'une chaîne | RIGHT | 55 |
| | code commun | 418 |

Fonctions Orhée

Le tableau suivant donne la consommation mémoire des fonctions Orphée.

| Fonctions Orphée | OF | Taille code |
|---|-------------|-------------|
| Déc. à gauche sur 32 avec récupération bits décalés | DSHL_RBIT | 152 |
| Déc. à droite sur 32 avec extension signe, récup. bits décalés | DSHR_RBIT | 152 |
| Déc. à droite sur 32 avec remplissage de 0, récup. bits décalés | DSHRZ_C | 133 |
| Déc. à gauche sur 16 avec récupération bits décalés | WSHL_RBIT | 91 |
| Déc. à droite sur 16 avec extension signe, récup. bits décalés | WSHR_RBIT | 103 |
| Déc. à droite sur 16 avec remplissage de 0, récup. bits décalés | WSHRZ_C | 90 |
| | code commun | 173 |
| Comptage/décomptage avec signalisation de dépassement | SCOUNT | 617 |
| Rotation vers la gauche d'un mot | ROLW | 41 |

| Fonctions Orphée | OF | Taille code |
|---|------|-------------|
| Rotation vers la droite d'un mot | RORW | |
| Rotation vers la gauche d'un double mot | ROLD | 49 |
| Rotation vers la gauche d'un double mot | RORD | 49 |

Fonctions de temporisation

Le tableau suivant donne la consommation mémoire des fonctions de temporisation..

| Fonctions temporisation | OF | Taille code (en mots) |
|--------------------------------|---------|-----------------------|
| Sortie créneau | FPULSOR | 215 |
| Temporisation de déclenchement | FTOF | 272 |
| Temporisation d'enclenchement | FTON | 217 |
| Temporisation d'impulsion | FTP | 245 |

Fonctions logarithmes, exponentielles et trigonométriques

Le tableau suivant donne la consommation mémoire des fonctions logarithmiques, exponentielles et trigonométriques..

| Fonctions logarithmes, exponentielles et trigonométriques | OF | Taille code (en mots) |
|---|-------------|-----------------------|
| Logarithme népérien | LN | 0 |
| Logarithme décimal | LOG | 0 |
| Exponentielle | EXP | 0 |
| Exponentiation d'un réel par un entier | EXPT | 523 |
| Partie entière | TRUNC | 128 |
| Cosinus d'un angle en radians | COS | 0 |
| Sinus d'un angle en radians | SIN | 0 |
| Tangente d'un angle en radians | TAN | 0 |
| Arc cosinus (résultat entre 0 et pi) | ACOS | 0 |
| Arc sinus (résultat entre -pi/2 et pi/2) | ASIN | 0 |
| Arc tangente (résultat entre -pi/2 et pi/2) | ATAN | 0 |
| Conversion degrés en radians | DEG_TO_RAD | 257 |
| Conversion radians en degrés | RAD_TO_DEG | 247 |
| | code commun | 392 |

Fonctions Régulation

Le tableau suivant donne la consommation mémoire des fonctions de régulation..

| Fonctions de régulation | OF | Taille code (en mots) |
|--|-------------|-----------------------|
| Régulateur PID mixte | PID | 1800 |
| Modulation en largeur d'impulsion d'une grandeur numérique | PWM | 600 |
| Etage de sortie de PID pour commande de vanne TOR | SERVO | 1200 |
| Gestion du dialogue opérateur dédié sur CCX17 des PID | PID_MMI | 4400 |
| | code commun | 573 |

Fonctions dialogue opérateur

Le tableau suivant donne la consommation mémoire des fonctions de dialogue opérateur..

| Fonctions Dialogue Opérateur | OF | Taille code (en mots) |
|--|----------------|-----------------------|
| Saisie bloquante d'une variable sur CCX17 | Ask_msg, | 46,5 |
| Saisie bloquante d'une variable sur msg contenu dans CCX17 | Ask_value, | 46,5 |
| Affectation dynamique des touches | Assign_keyS, | 46,5 |
| Commande de pilotage des voyants | Control_leds, | 46,5 |
| Affichage d'une alarme contenue dans le CCX17 | Display_alarm, | 46,5 |
| Affichage d'un groupe de messages contenus dans le CCX17 | Display_GRP, | 46,5 |
| Affichage d'un message contenu dans le CCX17 | Display_MSG, | 46,5 |
| Saisie multiple d'une variable sur CCX17 | GET_MSG, | 46,5 |
| Saisie multiple d'une variable sur msg contenu dans CCX17 | GET_VALUE, | 46,5 |
| Envoi d'une commande au CCX17 | PANEL_CMD, | 46,5 |
| Affichage d'un msg d'alarme contenu en mémoire automate | SEND_alarm, | 46,5 |
| Affichage d'un message contenu en mémoire automate | Send_msg | 46,5 |
| | code commun | 573 |

Fonctions de communication

Le tableau suivant donne la consommation mémoire des fonctions de communication..

| Fonctions Communication | OF | Taille code (en mots) |
|--|-------------|-----------------------|
| Lecture d'objets langage de base | READ_VAR | 617 |
| Ecriture d'objets langage de base | WRITE_VAR | 500 |
| Emission/réception de requêtes UNI-TE | SEND_REQ | 438 |
| Emission et/ou réception de données | DATA_EXCH | 375 |
| Emission d'une chaîne de caractères | PRINT_CHAR | 476 |
| Demande de lecture d'une chaîne de caractères | INPUT_CHAR | 625 |
| Emission et/ou réception d'une chaîne de caractères | OUT_IN_CHAR | 531 |
| Emission d'un télégramme | SEND_TLG | 219 |
| Réception d'un télégramme | RCV_TLG | 172 |
| Demande d'arrêt d'une fonction de communication en cours | CANCEL | |
| | code commun | 506 |
| Décalage d'1 octet vers la droite d'un tableau d'octets | ROR1_ARB | 235 |
| Serveur immédiat | SERVER | 32 |
| | code commun | 648 |
| Ecriture 1K de messagerie | | 32 |
| | code commun | 936 |
| Lecture 1K de messagerie | READ_ASYN | 32 |
| | code commun | 920 |

Fonctions commande de mouvement

Le tableau suivant donne la consommation mémoire des fonctions de commande de mouvement.

| Fonctions commande de mouvement | OF | Taille code (mots) |
|---|-------|--------------------|
| Commande de mouvement automatique 1 axe | SMOVE | 24 |
| Commande de mouvement automatique multi-axe | XMOVE | 32 |

Archivage des données

Le tableau suivant donne la consommation mémoire des fonctions d'archivage des données..

| Archivage de données | OF | Taille code (mots) |
|--|--------------|---------------------------|
| Initialisation zone d'archivage sur carte PCMCIA | SET_PCMCIA | 24 |
| Ecriture de données sur carte PCMCIA | WRITE_PCMCIA | 24 |
| Lecture de données sur carte PCMCIA | READ_PCMCIA | 24 |
| | code commun | 288 |

Fonction d'échange explicite

Le tableau suivant donne la consommation mémoire des fonctions de commande d'échange explicite..

| Echanges explicites | OF | Taille code (mots) |
|---|---------------|---------------------------|
| Lecture des paramètres de status | READ_STS | 0 |
| Lecture des paramètres de réglage | READ_PARAM | 0 |
| Mise à jour des paramètres de réglage | WRITE_PARAM | 0 |
| Sauvegarde des paramètres de réglage | SAVE_PARAM | 0 |
| Restitution des paramètres de réglage | RESTORE_PARAM | 0 |
| Mise à jour des paramètres de commande | WRITE_CMD | 0 |
| (1) OF spécifique, le code est compté dans le volume du module E/S. | | |

DFB de diagnostic

A la première programmation d'un des DFB de diagnostic, 200 mots sont réservés en zone programme de l'application.

Le tableau suivant indique pour chaque type de DFB de diagnostic la taille du code embarqué dans l'application (en zone programme) et la taille occupée par instance en zone donnée et en zone programme.

| DFB de diagnostic (tailles en mots) | Taille Type DFB | Taille code Type DFB | Taille données par instance |
|--|------------------------|-----------------------------|------------------------------------|
| IO_DIA | 800 | 64 | 72 |
| ALRM_DIA | 608 | 40 | 48 |
| NEPO_DIA | 15184 | 128 | 136 |
| TEPO_DIA | 10896 | 128 | 136 |
| EV_DIA | 1144 | 48 | 56 |
| MV_DIA | 2616 | 80 | 88 |
| ASI_DIA | 7912 | 304 | 312 |

Méthode de calcul du nombre d'instructions

Généralités

Cette méthode permet de calculer le nombre d'instructions de base (niveau assembleur) booléennes ou numériques.

Calcul du nombre d'instructions booléennes

Dans ce calcul sont pris en compte le nombre d'éléments suivants :

- opérations unitaires booléennes : chargement (LD), AND, OR, XOR, ST, ...
- parenthèses fermantes (ou les convergences ladder : liaisons verticales de convergence)
- blocs comparaison (AND[...], OR[...]) et operate ([...])

Ne pas compter comme instruction booléenne les opérateurs NOT, RE et FE.

Exemple :

```
LD %M0
AND ( %M1
OR %M2
)
ST %M3
= 5 instructions booléennes
```

Calcul du nombre d'instructions Numériques

Dans ce calcul sont pris en compte le nombre d'éléments suivants :

- affectations (:=)
- chargement de la première valeur après :=
- instructions arithmétiques (+, -, *, /, <, =, ...), opérations sur mots ou tableaux de mots, doubles mots, flottants)
- instructions logiques sur mots
- fonctions (OF, EQUAL, ...) quel que soit le nombre de paramètres
- blocs fonction (ou instruction de bloc fonction)

Exemple :

```
%MW0 := ( %MW1 + %MW2 ) * %MW3 ;
```

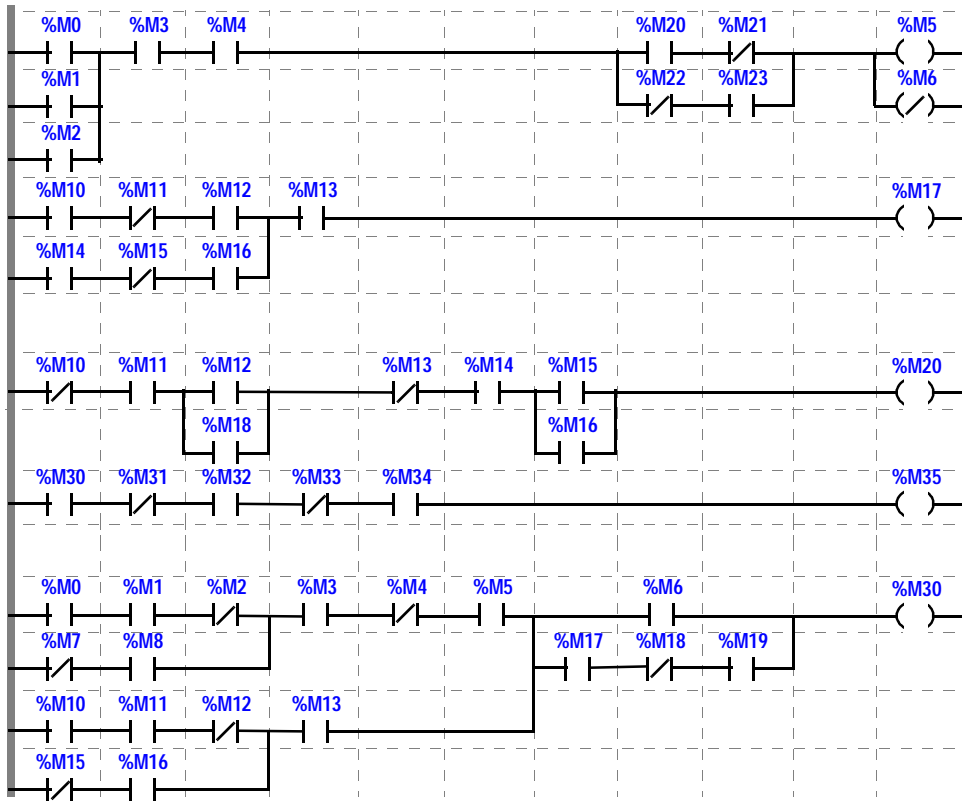
instructions comptabilisées :

```
:=
%MW1 (correspond à l'instruction de chargement dans l'accumulateur )
+
*
```

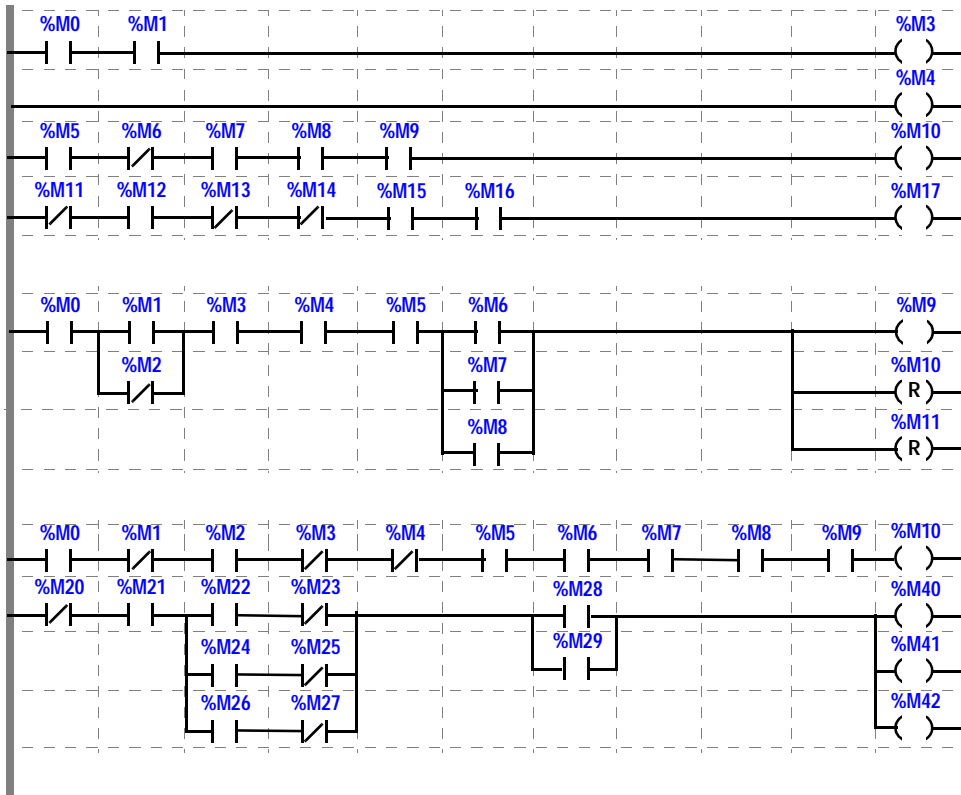
soit 4 instructions.

Exemple

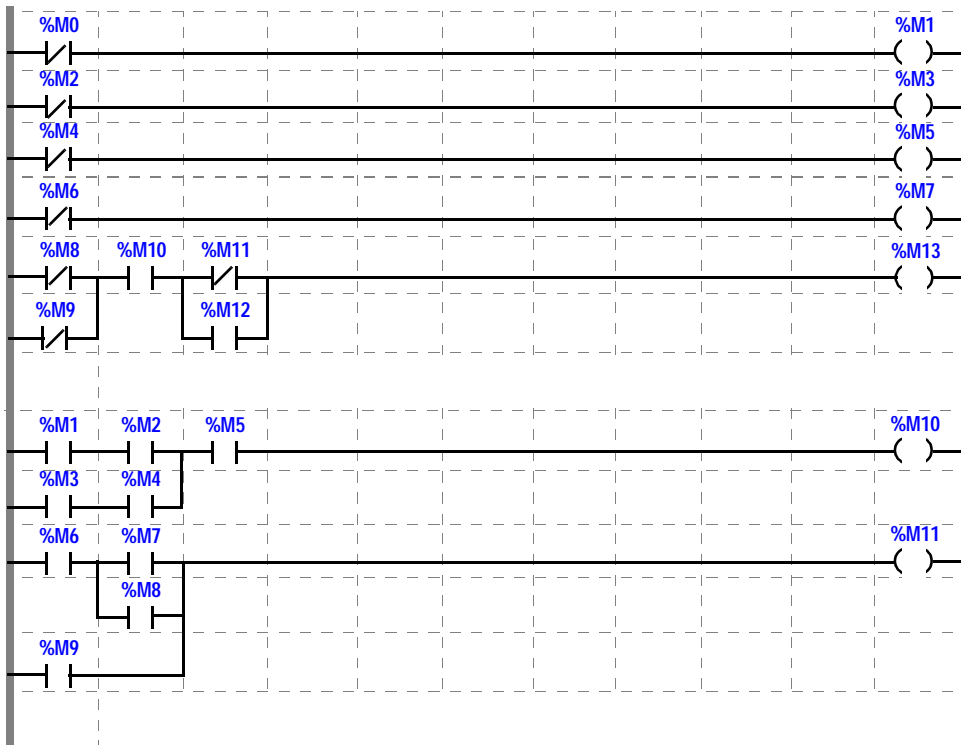
Exemple de programme comportant 65% de booléen et 35% de numérique :



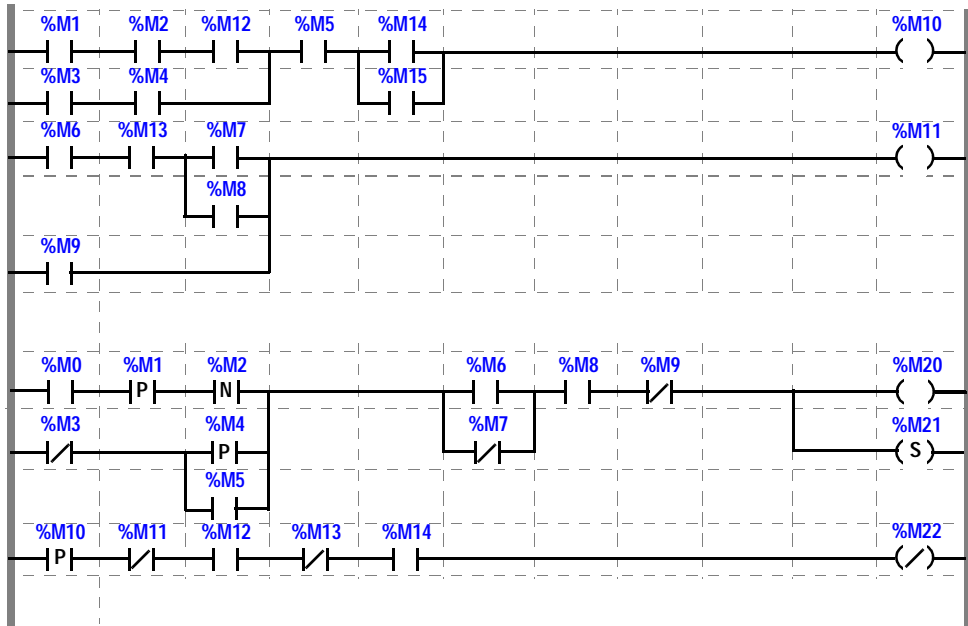
Exemple (suite)



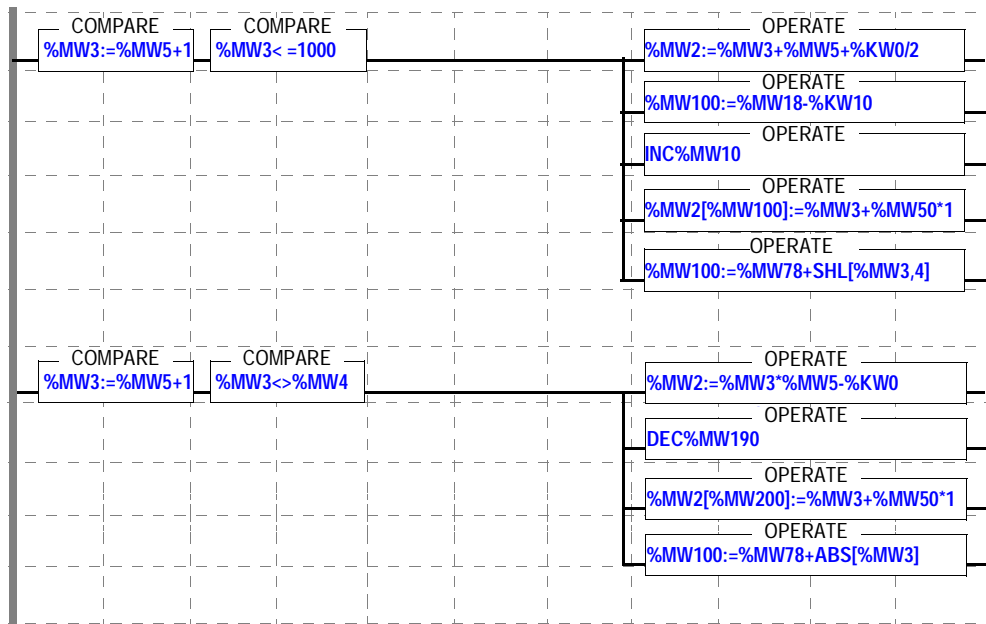
Exemple (suite)



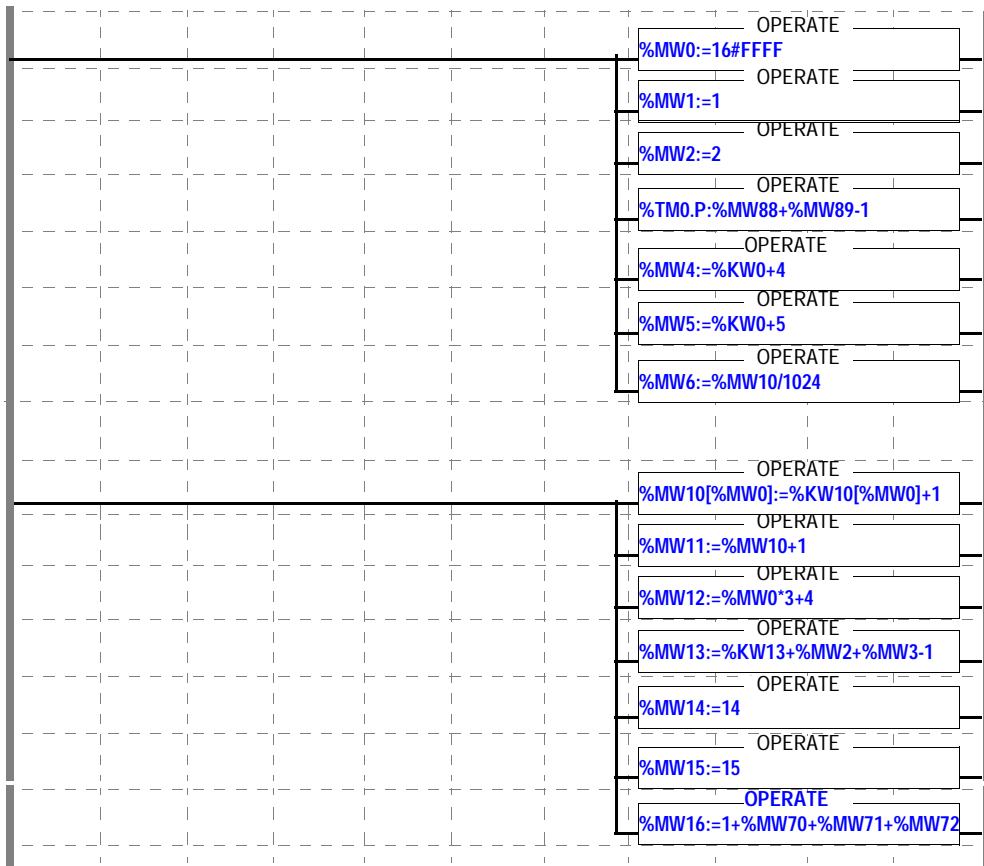
Exemple (suite)



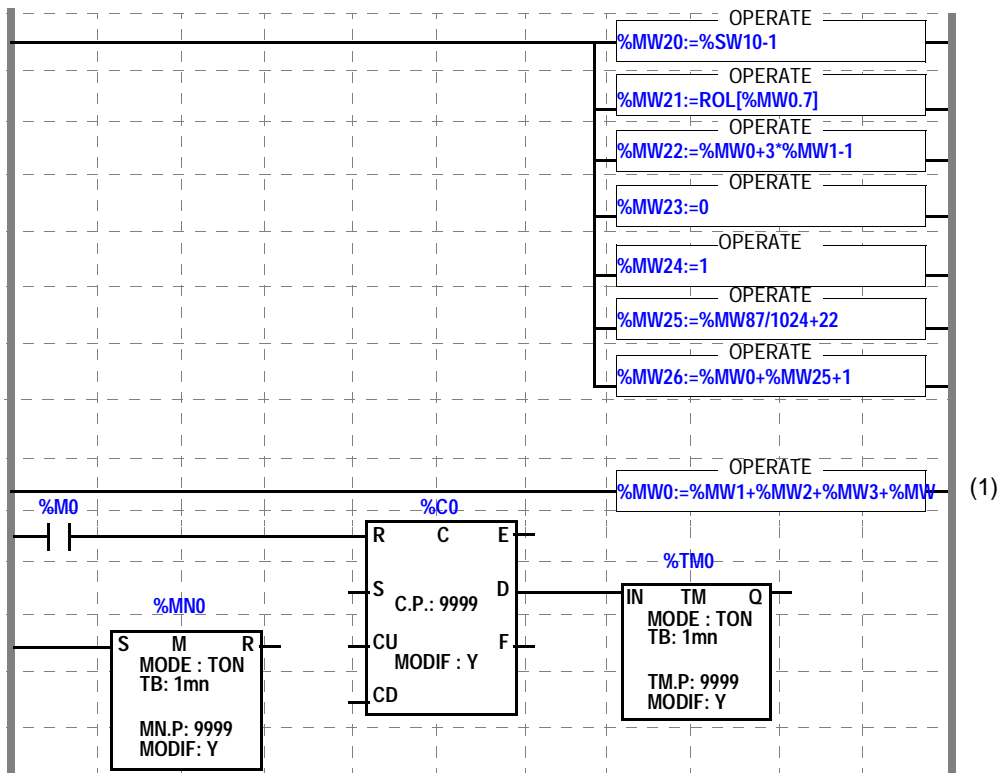
Exemple (suite)



Exemple (suite)



Exemple (suite)

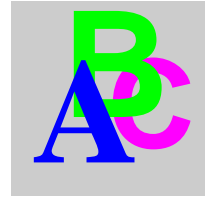


(1):%MW0:=%MW1+%MW2+%MW3+%MW4+%MW5+%MW6+%MW7+%MW8+%MW9+%MW10+1

Bilan

| | Nombre d'instructions | % | |
|-------------------------------------|-----------------------|--------|---------|
| Booléen sans front | 187 | 54,05% | 64,16% |
| Booléen avec front | 4 | 1,16% | |
| Bloc opération | 31 | 8,96% | |
| Bloc fonction | 3 | 0,87% | 35,84% |
| Arithmétique simple (+,-,=,AND,...) | 111 | 32,08% | |
| Arithmétique indexée | 4 | 1,16% | |
| *, / | 6 | 1,73% | |
| Valeurs immédiates | 24 | | |
| Total | 346 | | 100,00% |

Index



C

CEI1131-3, 56
CloseDFB, 121
CloseIOModule, 119
CloseProgram, 117
CloseStx, 88
ConnectPLC, 94

D

DisconnectPLC, 93
DownloadDataToPLC, 96
DownloadToPLC, 98

E

ExportFefFileWithConfiguration, 92
ExportFeFile, 91
ExportScyFile, 90

F

Fonctions OLE, 85

G

GetMessageError, 122
GetPL7State, 107
GetPLCApplIdentity, 109
GetServerVersion, 123
GetSTXApplIdentity, 108

GetSymbol, 104

I

ImportScyFile, 124

O

OLE, 77
OpenStx, 87
OpenStxWithMaj, 89
OpenTool, 113

S

SaveStx, 95
SendCommandToPLC, 111
SetDownloadFunction, 106
SetDriverAndAdresse, 112
SetPosPL7Windows, 115
SetServerIHM, 105
ShowDFB, 120
ShowIOModule, 118
ShowProgram, 116

U

UploadDataFromPLC, 100
UploadFromPLC, 99
UploadFromPLCWithSymbol, 102

