

Architecture XWAY de Schneider

Armand TOGUYENI

Professeur des Universités

Centrale Lille Institut,

B.P. 48, 59651 Villeneuve d'Ascq Cedex,

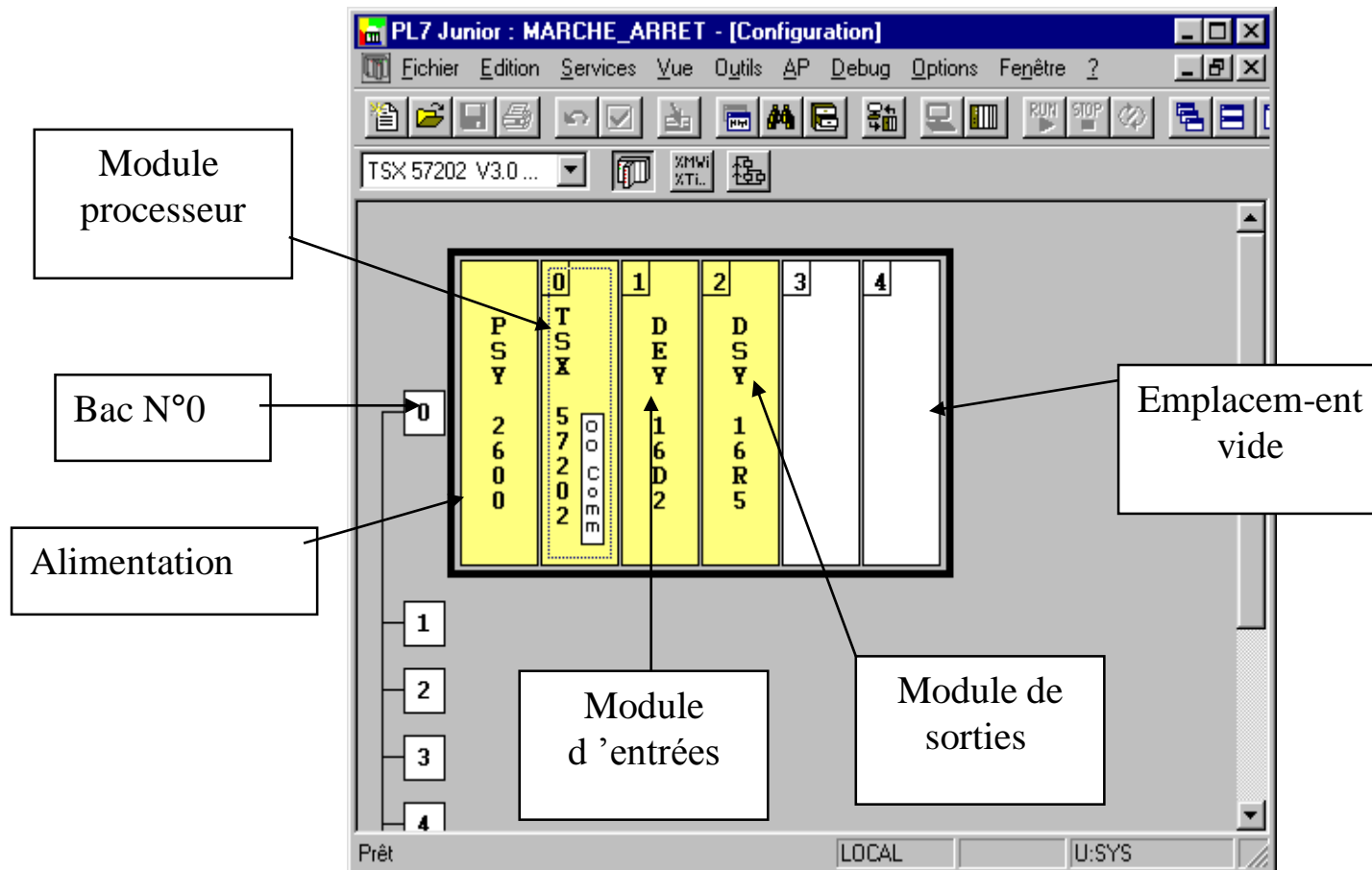
FRANCE

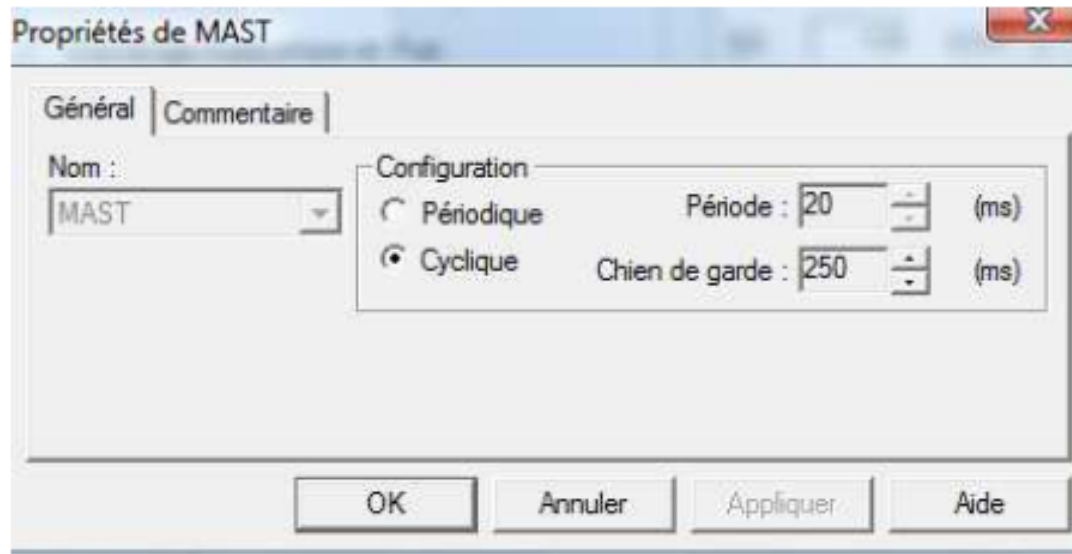
Tel: (33 / 0)3 20 33 54 49, Fax: (33 / 0)3 20 33 54 18

mel : armand.toguyeni@centralelille.fr

- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- **Exemples de codage de requête**
- **Encapsulation dans des paquets Modbus TCP/IP**

- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- **Exemples de codage de requête**
- **Encapsulation dans des paquets Modbus TCP/IP**





La tâche peut être périodique c'est-à-dire s'exécuter après le temps défini

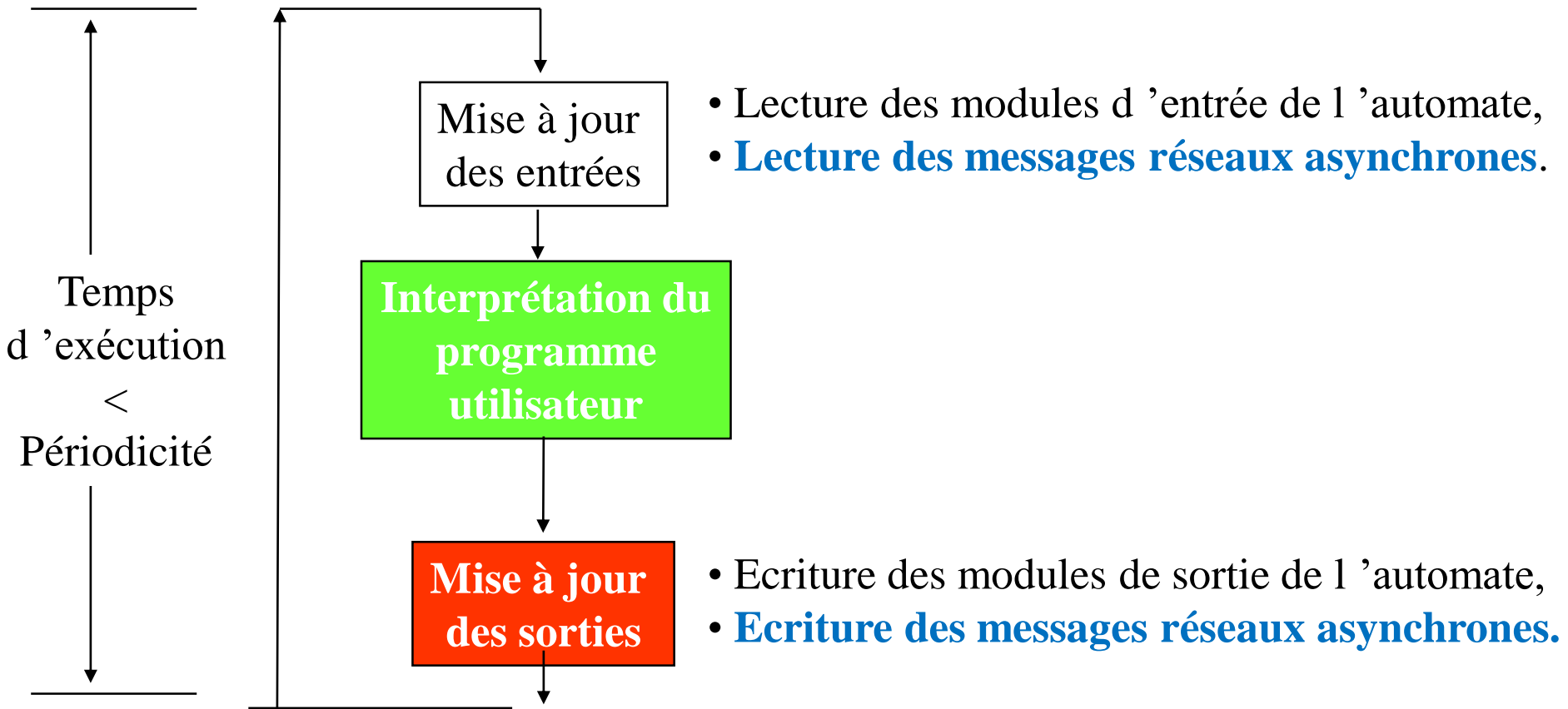
Ou cyclique (ce n'est valable que pour la tâche maître), c'est-à-dire s'exécuter dès qu'elle est finie

On peut définir la valeur du chien de garde

La tâche maître doit être défini périodique si vous utiliser une tâche auxiliaire

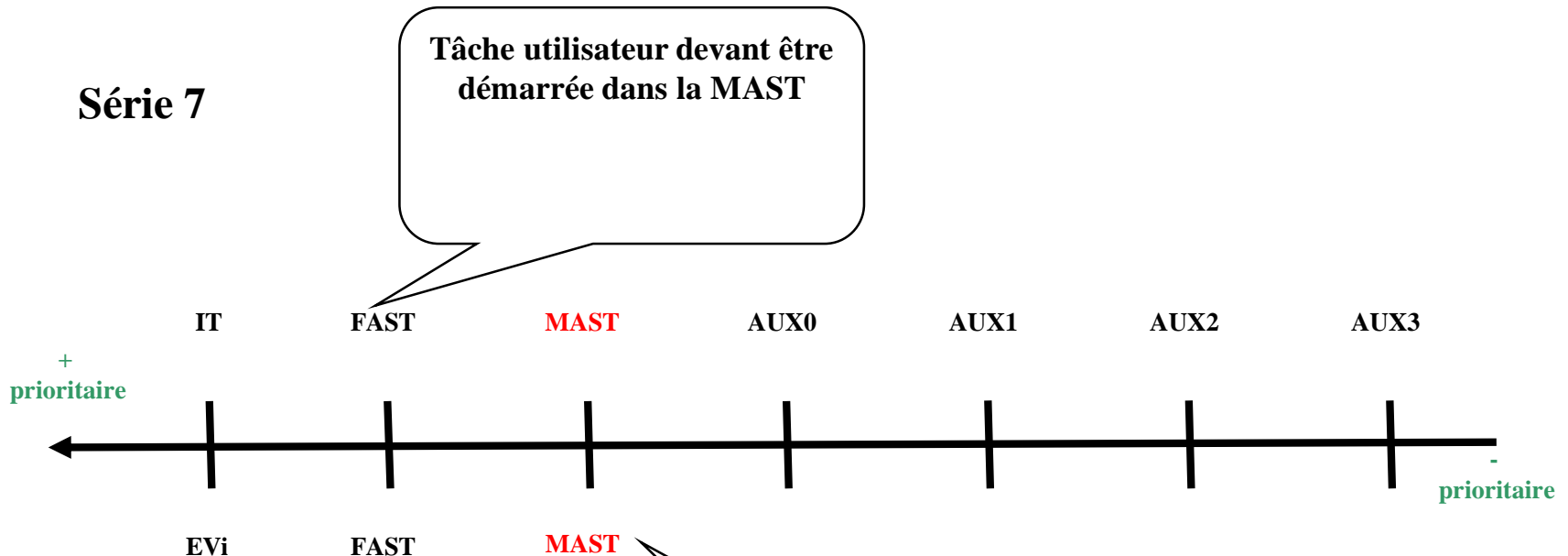
- OS de Schneider : Fonctionnement multitâches orchestré par le moniteur automate (tâche système démarrant les tâches « utilisateur »)
- Deux types de tâches utilisateur :
 - Tâches asynchrones (périodiques) : caractérisée par une périodicité de déclenchement appelée temps de cycle,
 - Tâche principale : **MAST** (tâche maître, tâche par défaut), temps de cycle moyen = 50 ms,
 - Tâche rapide : **FAST**, temps de cycle moyen = 10 ms.
 - Tâches auxiliaires (Uniquement la série 7) : **AUX0 ... AUX3**
 - Tâches événementielles : déclenchées par des interruptions matérielles issues de modules métiers comme les coupleurs réseaux, commande d'axes.
 - **IT** (spécifique à la série 7): une seule tâche d'interruption
 - **EV_i** ($i \in [0 ..MAX]$): spécifique aux **PREMIUM**

Fonctionnement de type **Asynchrone** (par rapport aux E/S)



Un seul processeur automate => une seule tâche active à la fois

Série 7



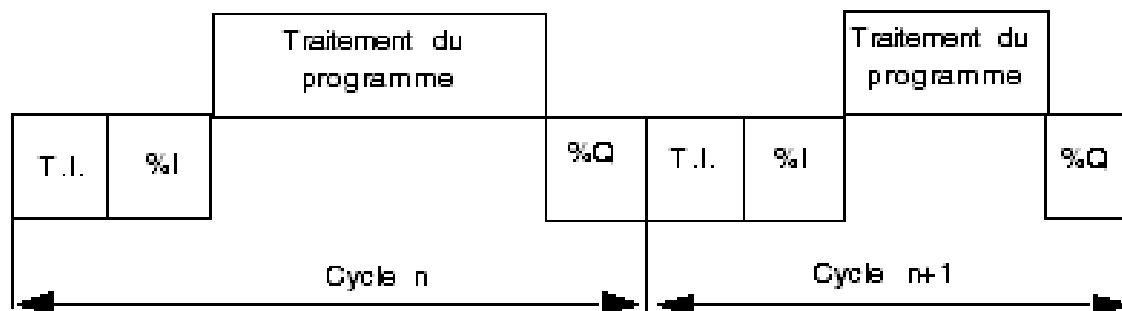
Prémium

Seule tâche utilisateur active par défaut

- Cas des applications simples monotâches
 - Seule la tâche MAST est active
 - L'utilisateur a le choix :
 - Fonctionnement cyclique,
 - Fonctionnement périodique.
- Application multitâches
 - Les tâches asynchrones doivent obligatoirement être configurées en périodique
 - Plus une tâche est prioritaire, plus sa période doit être faible

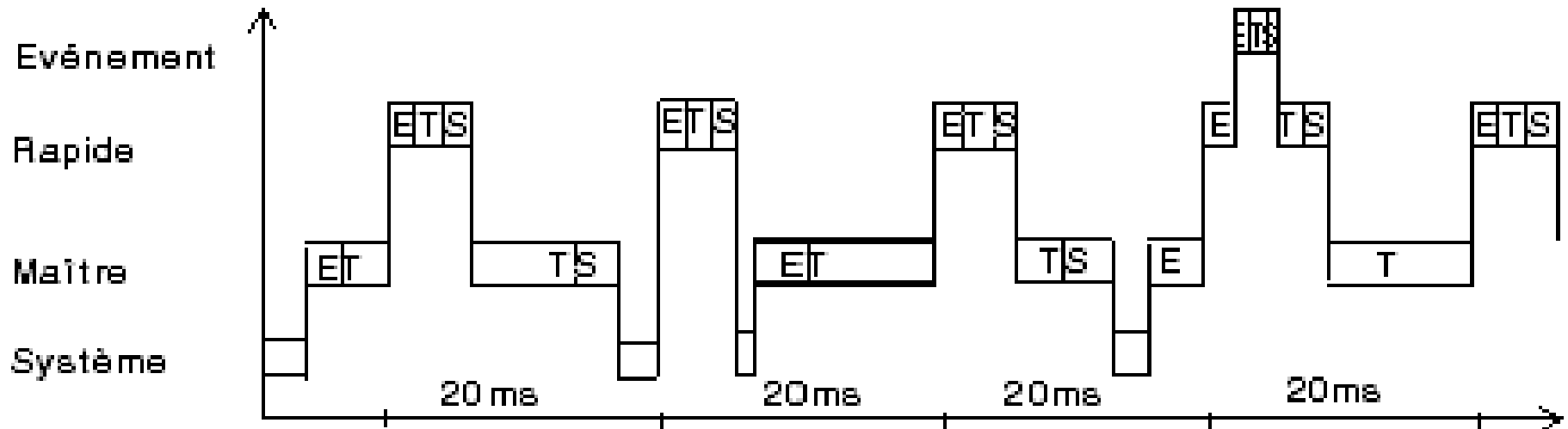
Architecture multitâches (6) : fonctionnement cyclique

- T.I (Traitement interne): Le système réalise implicitement la surveillance de l'automate (gestion des bits et mots système, mise à jour des valeurs courantes de l'horodateur, mise à jour des voyants d'état, détection des passages RUN/STOP, ...) et le traitement des requêtes en provenance du terminal (modifications et animation).
- %I (Acquisition des entrées)
- Traitement du programme utilisateur
- %Q (Mise à jour des sorties)

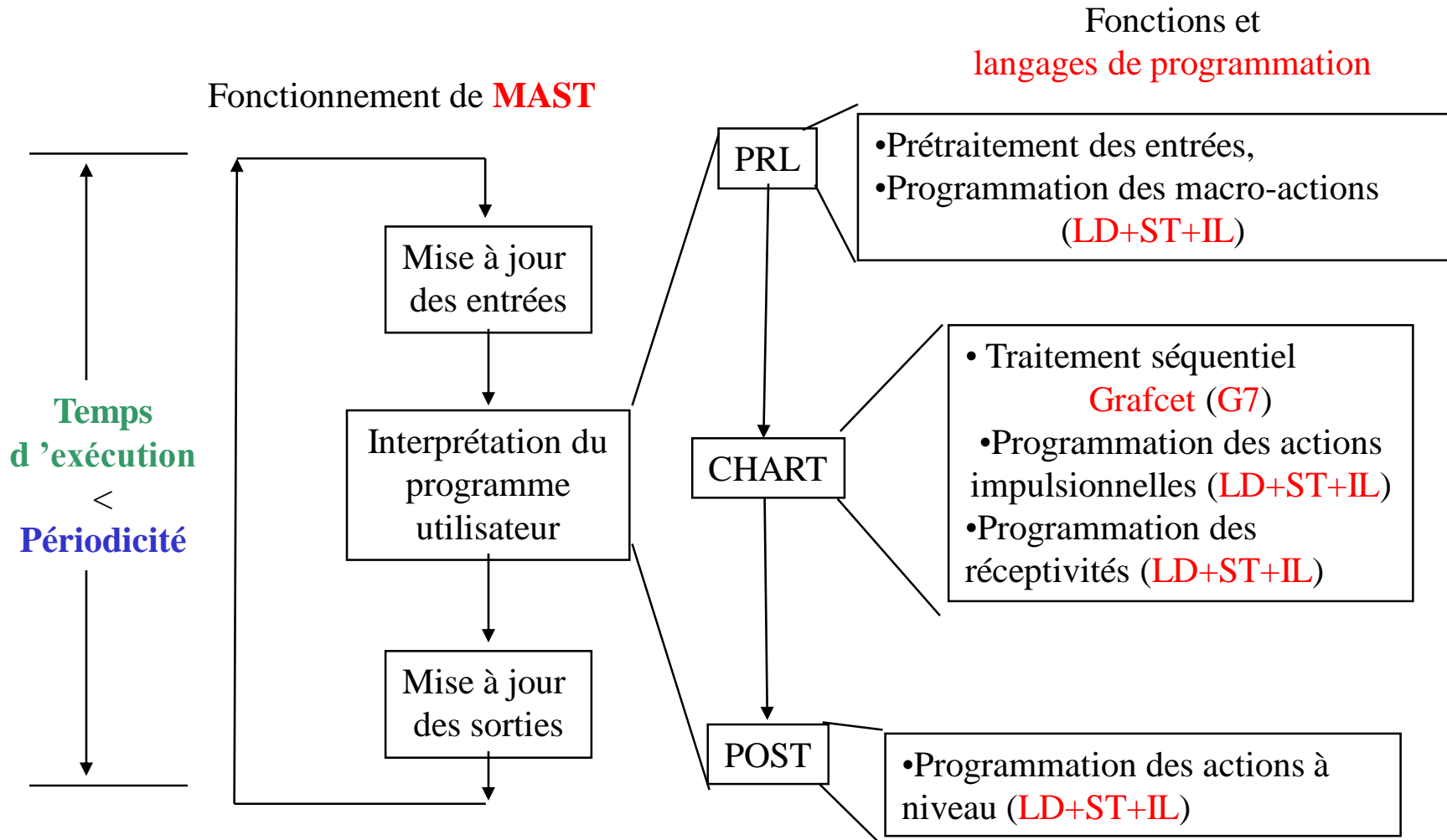


Architecture multitâches (7) : fonctionnement périodique

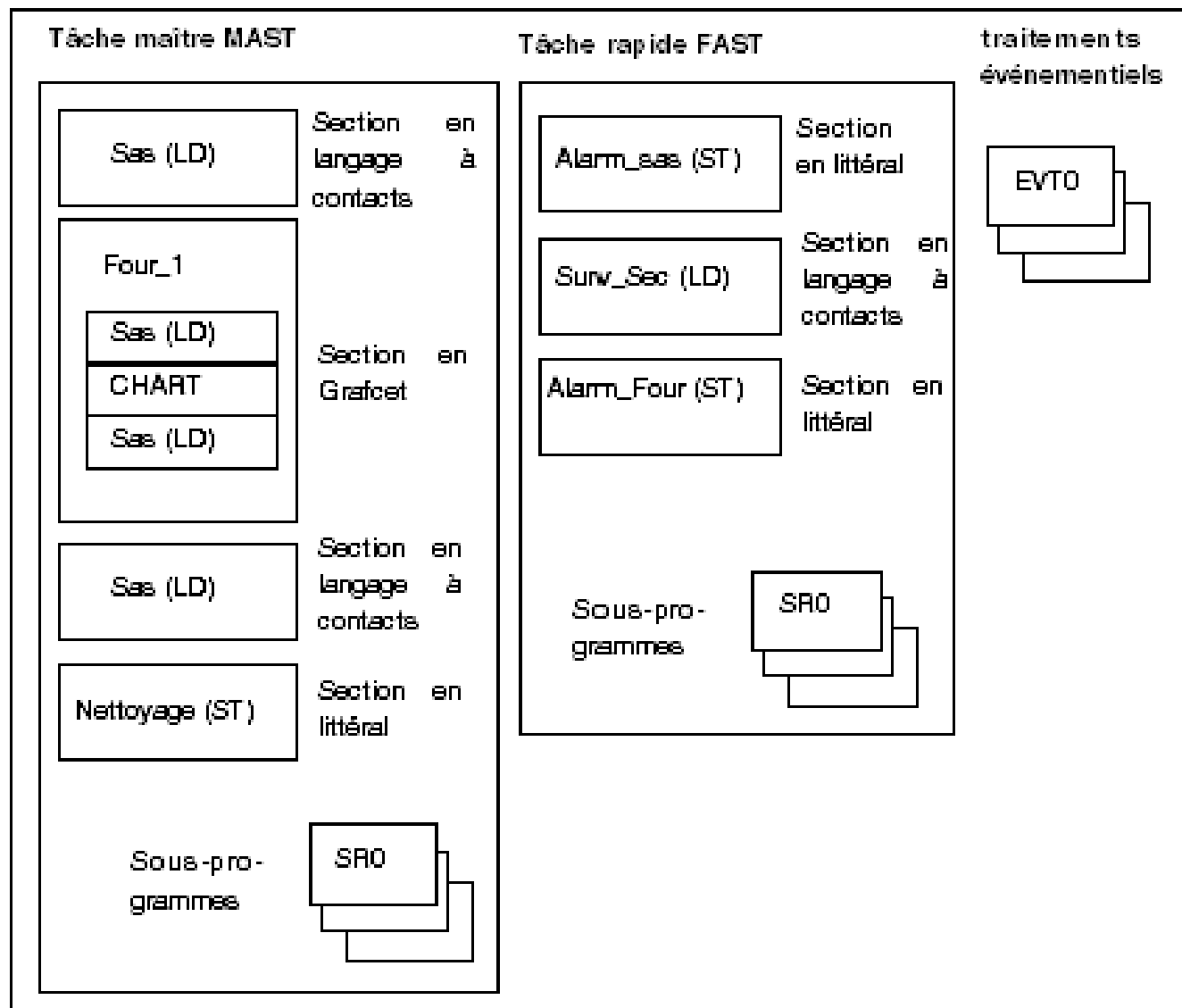
- E : acquisition des entrées
- T : traitement du programme
- S : mise à jour des sorties



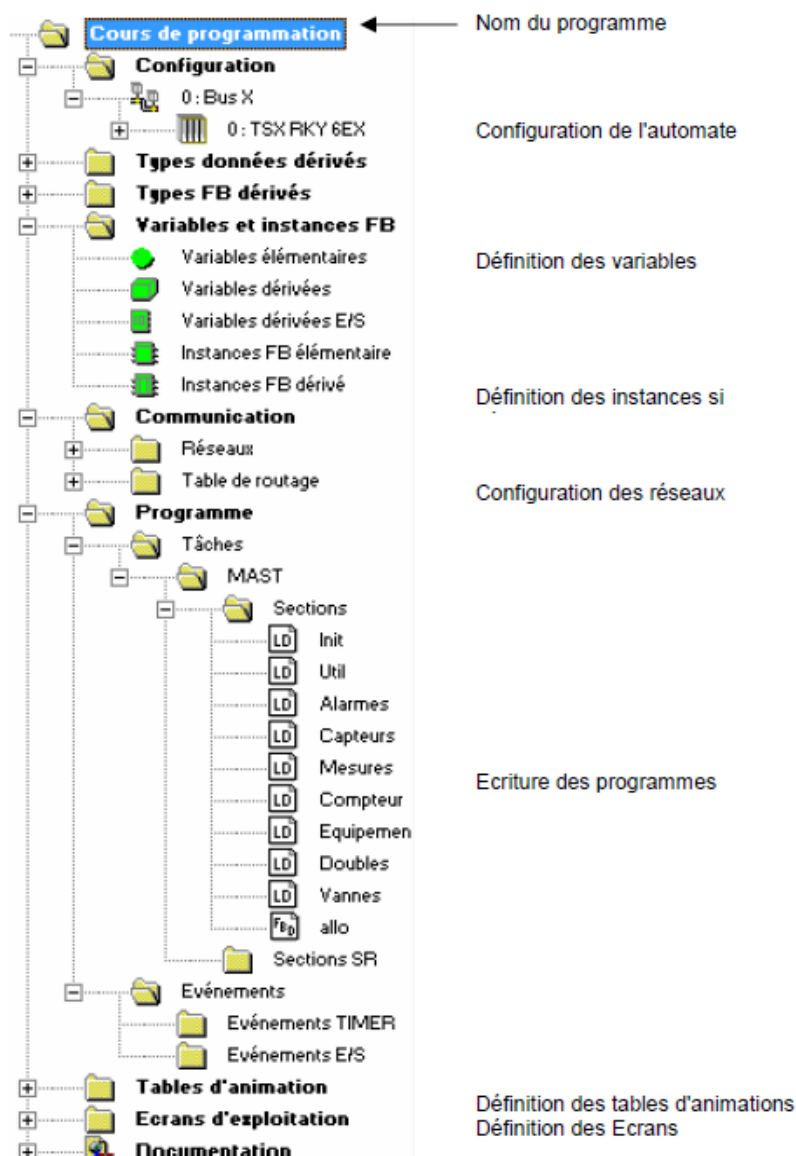
Système multi-langages : programmation de la tâche MAST



Structure logicielle PL7



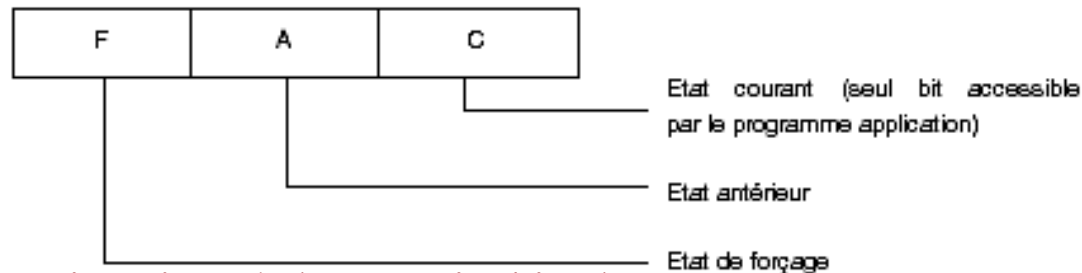
Structure logicielle Unity



- La mémoire RAM d'un automate est composé de mots (16 bits)
- La mémoire se décompose en trois segments :
 - **le programme** : descripteurs et code exécutable des tâches,
 - **les données** : données dynamiques de l'application et données système (le système réserve une zone mémoire RAM de 5 Kmots minimum)
 - **les constantes** : mots constants, valeurs initiales et configuration des entrées/sorties.

- Le segment données se décompose en :
 - **Booléens ou bits (4096 bits maximum) :**
 - **E/S :**
 - %I3.4.12 : entrée 12 du module 4 du bac 3
 - %Q6.10.7 : sortie 7 du module 10 du bac 6
 - **Bits internes : %Mi avec $i \in [0 .. MAX]$**
 - MAX (512 par défaut) est défini par configuration
 - **Bits systèmes : %Si avec $i \in [0 .. 63]$**
 - %S21 : Mis à 1 dans le PRL provoque la réinitialisation de tous les grafjets de l'API,
 - %S22 : Mis à 1 dans le PRL provoque la désactivation de tous les grafjets de l'API,
 - %S19 : mis à 1 par le système pour signaler un débordement de tâche.
 - **Bits d'activation : S_i_j.x (Etape j de la section S_i)**

- Le codage des objets bits EBOOL permet le test de front montant ou descendant sur : les bits d'entrées/sorties, les bits internes.



- Lors de la mise à jour de la mémoire bits, le système assure :
 - Etape 1 : le transfert de l'image de l'état courant dans l'état antérieur.
 - Etape 2 : la réactualisation de l'état courant par le programme, le système ou le terminal (cas du forçage d'un bit).

- Le segment données (suite) :
 - Mots (association de 16 bits) :
 - Mots internes : $\%MW_i$ avec $i \in [0 .. MAX]$
 - MAX (1024 par défaut) est défini par configuration
 - $\%MW_i = 10$ (décimal),
 - $\%MW_i = 16\#000A$ (hexadécimal),
 - $\%MW_i = 2\#0000\ 0000\ 0000\ 1010$ (binaire),
 - Octets internes : $\%MB_i$
 - Doubles mots internes : $\%MD_i$
 - Mots systèmes : $\%SW_i$
 - Mots de temporisation d'étape : $S_{i_j.t}$

	Poids fort	Pds faible	
$\%MW_{max}$			
$\%MW_{(i+1)}$			
$\%MW_i$	$\%MB(2i+1)$	$\%MB(2i)$	$\%MD_i$
$\%MW_0$			

Segments mémoire (5) : constantes et données complexes

- **Constantes :**
 - **Mots constantes : %KW_i**
 - **Octets constants : %KB_i**
- **Blocs fonctions :**
 - **Sont composés de bits, mots, et constantes qui occupent une partie des espaces logiques correspondants :**
 - **Différents types :**
 - **Timer série 7 (Utilisé en séance de TP 2)**
 - **Registre (LIFO ou FIFO ; utilisé en L4)**
 - **Monostable,**
 - **Compteurs,**
 - **...**

Actions associées aux étapes

Propriétés Etape : etap2

Général Actions Commentaire

Qualificatif Temps Action

☒ Valeur ☐ Variable ☒ Variable ☐ Section

P1

et 1

Qualificatif	Action
P1	et 1
N	eta 1

Modifier

Nouveau

Supprimer

Vers le haut

Vers le bas

Initialiser la recherche pour la variable sélectionnée...

OK Annuler Appliquer Aide

Qualificatifs

Qualificatif

P1

None

N

R

S

L

D

P

DS

P1

P0

N : Normal

R : Raz l'action précédente

S : Mémoire l'action qui sera désactivés par une étape "R"

L : Action temporisé, le temps est défini ou dans une variable

D : Action retardé, le temps est défini ou dans une variable

P : Action ne dure d'un tour de scrutation

DS: Action retardé puis mémorisé, désactivé par une étape "R"

P1: Action sur front montant, s'exécute en premier, "Equivalent à l'activation du PL7"

P0: Action sur front descendant, s'exécute en dernier "Equivalent à la désactivation du PL7"

Propriétés des transitions

Condition de transition | Commentaire

☐ Inverser condition de transition

Type de condition de transition

☐ Section TRANSITION ☒ Variable

Variable BOOLEENNE, valeur ou repère

tu1

OK Annuler Appliquer Aide

Options du projet

Accessible depuis "Outils"

Options du projet

Génération | Editeurs | Extensions de langage | Ecrans d'exploitation

Initialisation

Éléments de structure

- ☒ Autoriser les procédures
- ☒ Autoriser les sous-programmes

Diagramme fonctionnel en séquence (SFC)

- ☒ Autoriser les macros
- ☐ Autoriser plusieurs jetons

Types de données

- ☒ Autoriser l'utilisation du front sur EBOOL
- ☒ Autoriser INT/DINT à la place de ANY_BIT
- ☒ Autoriser l'extraction de bits pour INT & WORD
- ☐ Autoriser la représentation directe de tableaux (%xxx.nn)
- ☐ Autoriser les tableaux dynamiques [ANY_ARRAY_XXX]
- ☐ Inhiber le contrôle de taille des tableaux

Identificateurs

- ☐ Chiffres en début autorisés
- ☐ Jeu étendu de caractères autorisé

Langages graphiques (FBD/LD)

- ☒ Utilisation d'expressions ST

Langages textuels (IL/ST)

- ☒ Autoriser les paramètres vides dans les appels informels
- ☐ Autoriser les sauts et les labels (ST)
- ☐ Autoriser les affectations en cascade [a:=b:=c:] (ST)
- ☐ Autoriser les commentaires imbriqués

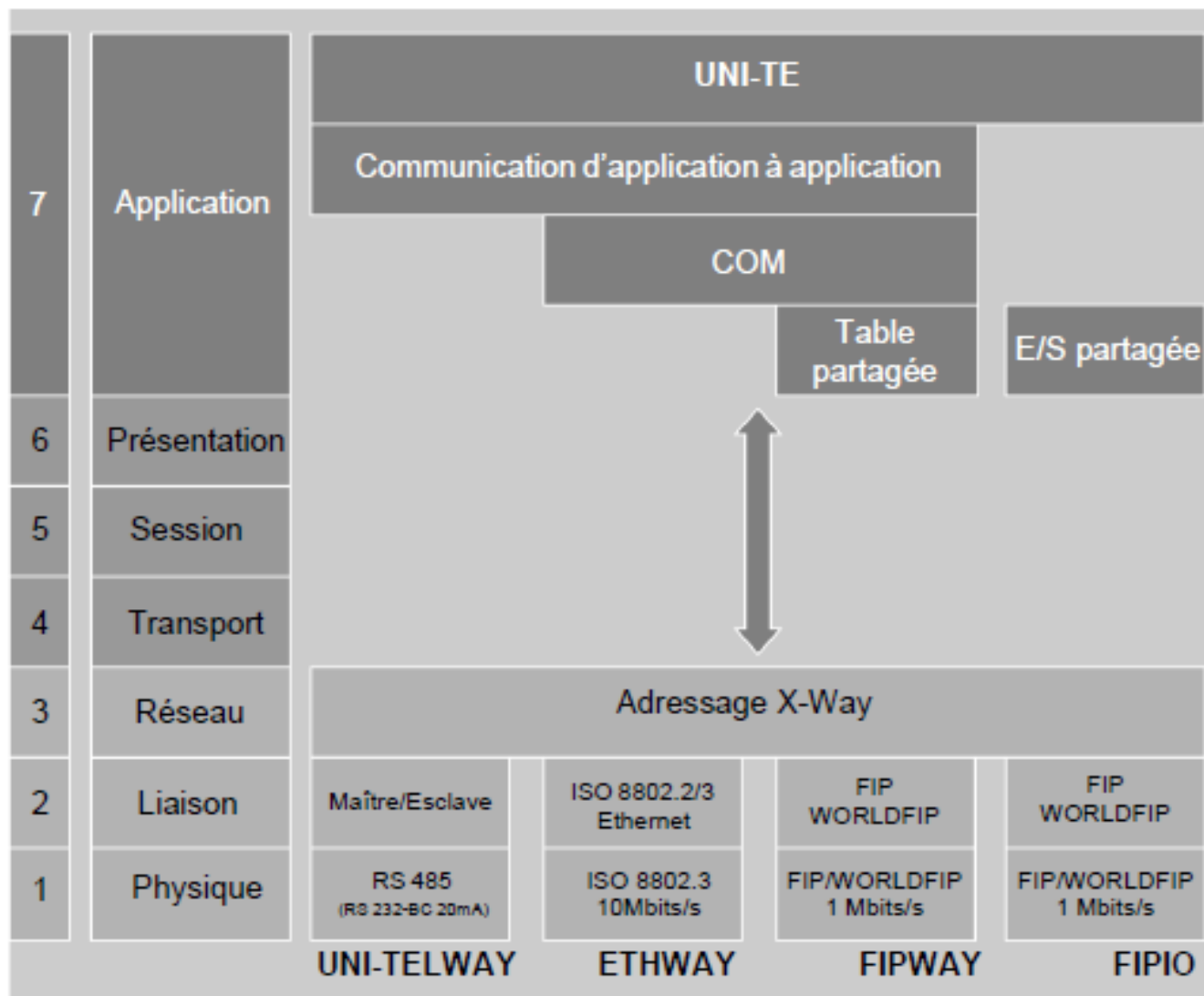
OK Annuler Aide

- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- **Exemples de codage de requête**
- **Encapsulation dans des paquets Modbus TCP/IP**

Architecture XWAY(1) : L'OFFRE DE SCHNEIDER

- ❑ Bus de terrain : FIPIO,
- ❑ Bus de terrain : UNI-TELWAY,
- ❑ réseau TELWAY,
- ❑ réseau FIPWAY,
- ❑ réseau ETHWAY,
- ❑ réseau MAPWAY,
- ❑ réseau OSI ETHERNET.

Architecture XWAY (2)



Architecture XWAY (3)

Application				Télégrammes			
			Mots communs				
	Messagerie industrielle UNITE						Messagerie MMS
Présentation							
Session							
Transport							TP4
Routage	Routage XWAY						
Liaison	UNITELWAY (Maitre/Esclaves)	FIPIO (Arbitre de bus)	TELWAY7 (Arbitre de bus)	FIPWAY (Arbitre de bus)	MAPWAY (IEEE 802.4 ; Jeton sur Bus)	ETHWAY (IEEE 802.3 ; CSMA/CD)	ETHERNET/MMS (IEEE 802.3 ; CSMA/CD)
Physique	Bus ; D=9600 b/s	Bus ; D=1 Mb/s	Bus ; D=19200 b/s	Bus ; D=1Mb/s	Bus ; D=5Mb/s	Bus ; D=10 Gb/s	Bus ; D=10 Gb/s
	Bus de terrain		Réseaux				

- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- **Exemples de codage de requête**
- **Encapsulation dans des paquets Modbus TCP/IP**

Architecture XWAY (3) : adressage

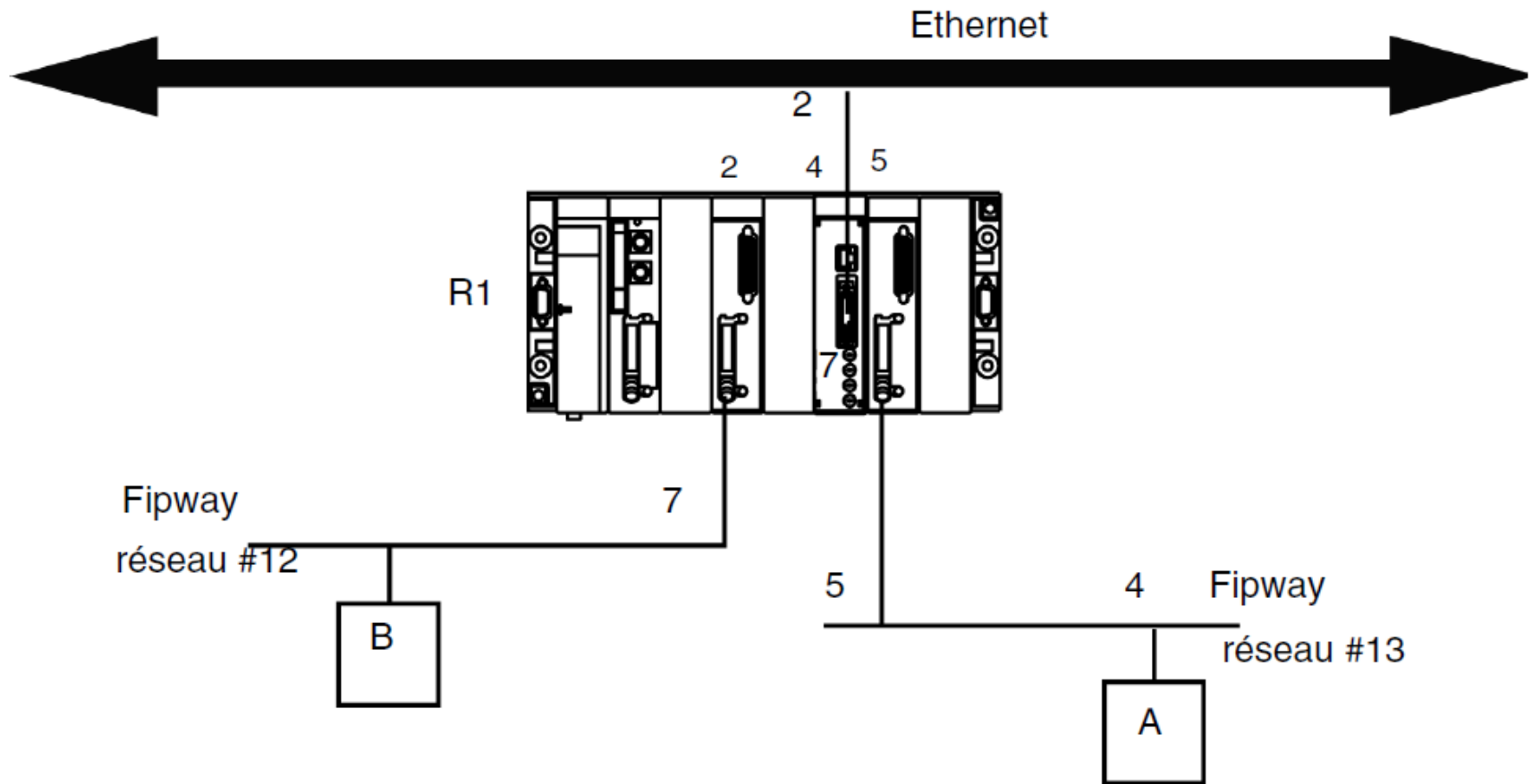
- ❑ **Adressage simple sur 2 niveaux** : Permet l'adressage d'une station dans une architecture mono-réseau ou multi-réseaux.
 - ◆ **N° réseau** : 0 pour une architecture mono-réseau sinon de 1 à 127
 - ◆ **N° station** : 254 pour une station isolée sinon de 1 à 100
- ❑ **Adressage standard sur 5 niveaux** : Permet d'adresser des équipements se trouvant sur un bus. 3 niveaux supplémentaires :

Architecture XWAY (4) : adressage simple

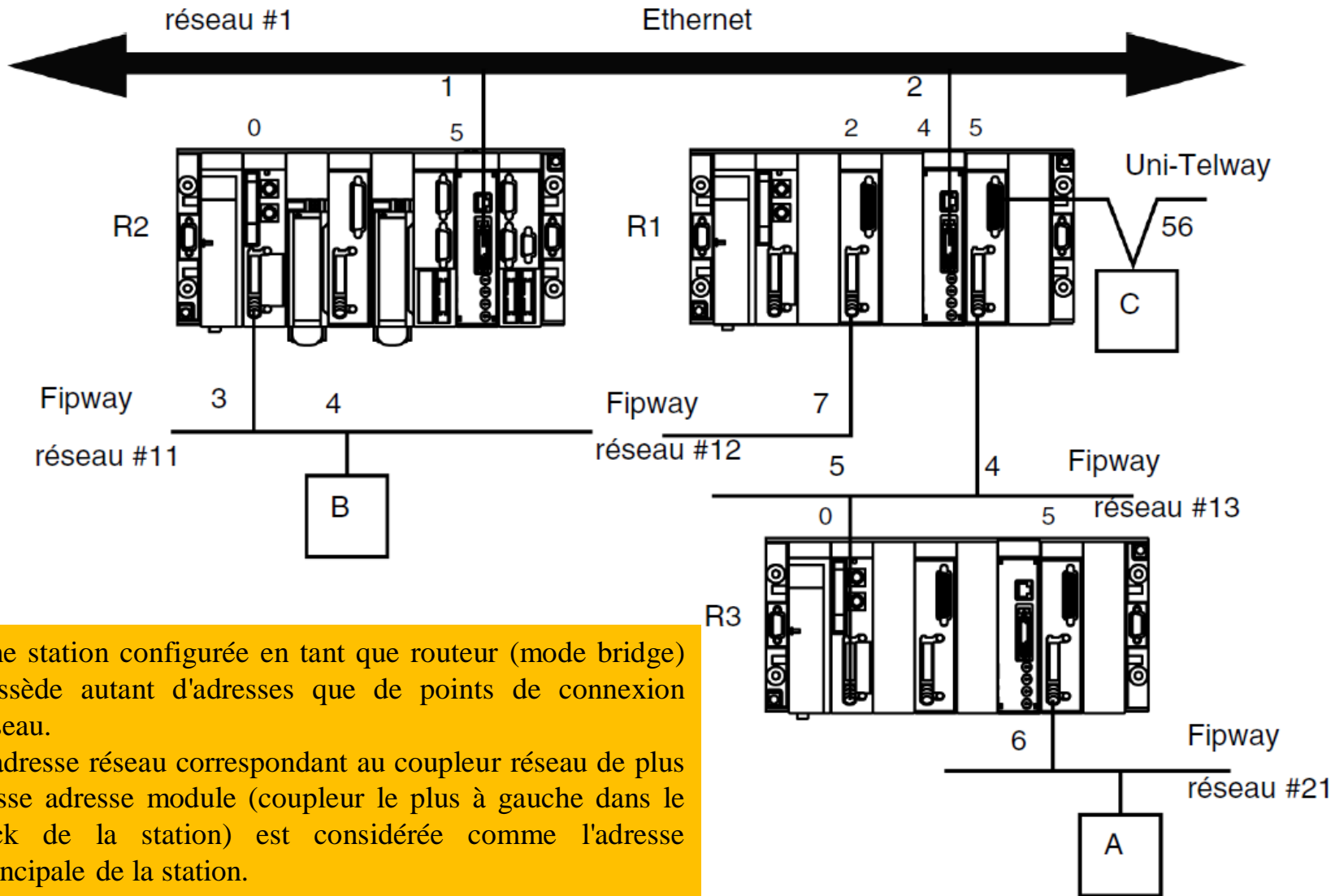
- ❑ Un équipement sur le réseau de cellule FIPWAY est identifié par une adresse unique formée du numéro de réseau et du numéro de station.
- ❑ Numéro de réseau
 - ♦ 0 dans une architecture monoréseau
 - ♦ 1 à 127 dans une architecture multiréseaux
- ❑ Numéro de station
 - ♦ C'est l'adresse physique de la station . Elle est comprise entre 0 et 63.
- ❑ Extension d 'adresse :
 - ♦ **APP** : désigne l 'application de la station destinatrice,
 - ♦ **SYS** : désigne une fonction système,
 - ♦ **ALL** : diffusion.

- ❑ Numéro de réseau (r)
 - ◆ 0 dans une architecture monoréseau
 - ◆ 1 à 127 dans une architecture multiréseaux
- ❑ Numéro de station (s)
 - ◆ C'est l'adresse physique de la station . Elle est comprise entre 0 et 63.
- ❑ 4 niveaux supplémentaires pour un bus unitelway :
 - ◆ Bac (b)
 - ◆ Module (m)
 - ◆ N° voie (v) : 0 ou 1
 - ◆ N° équipement sur le bus € : adresse liaison de l'équipement
 - ◆ Exemple : {r.s}b.m.v.e

Architecture XWAY (6) : Adressage d'une station multi-coupleurs

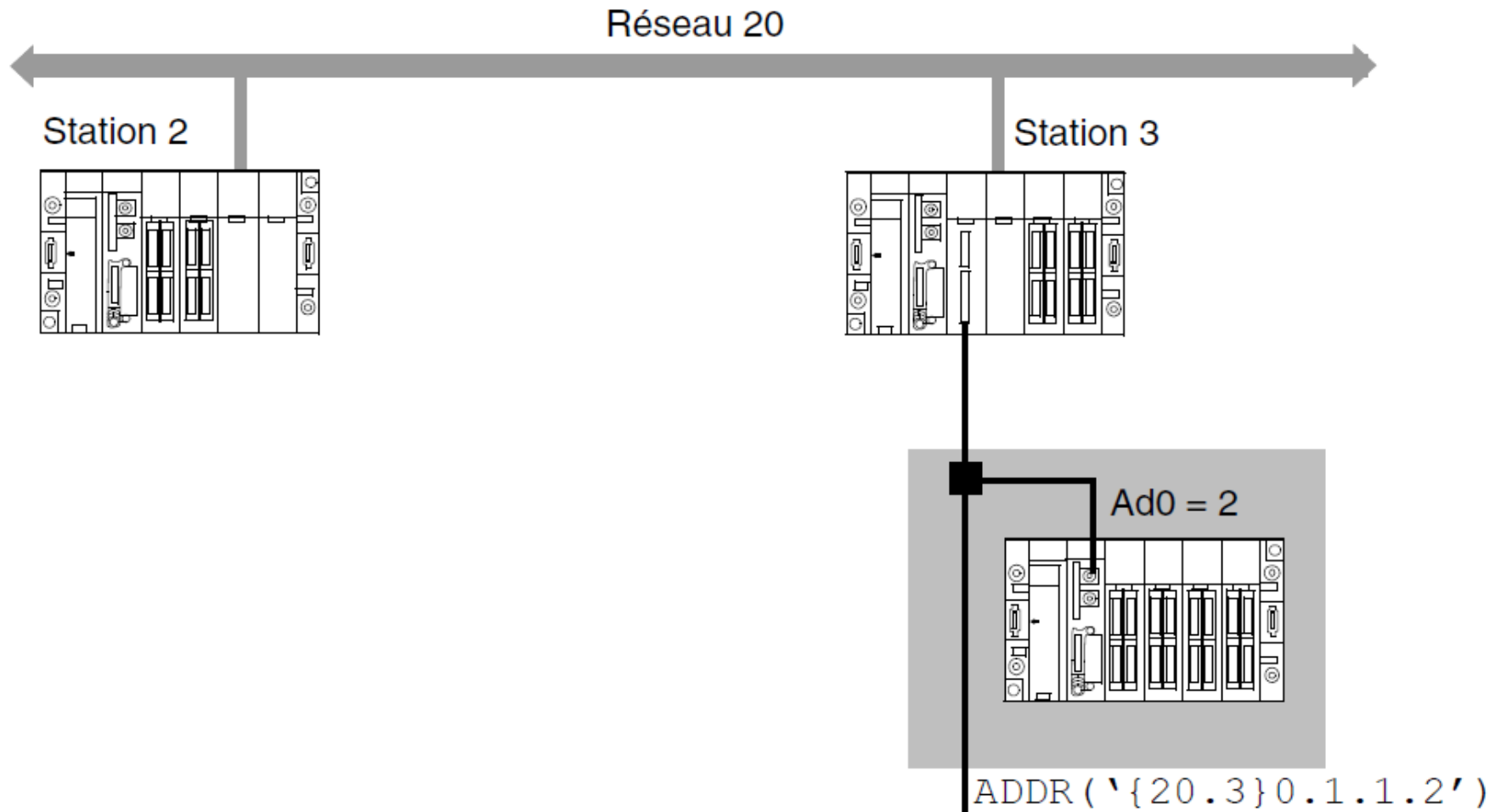


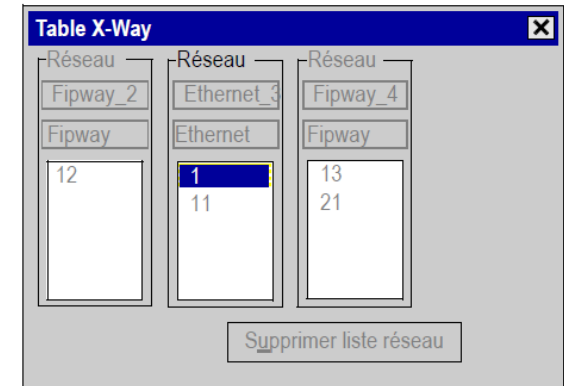
Architecture XWAY (7) : Adressage d'une station en mode routeur



Une station configurée en tant que routeur (mode bridge) possède autant d'adresses que de points de connexion réseau.

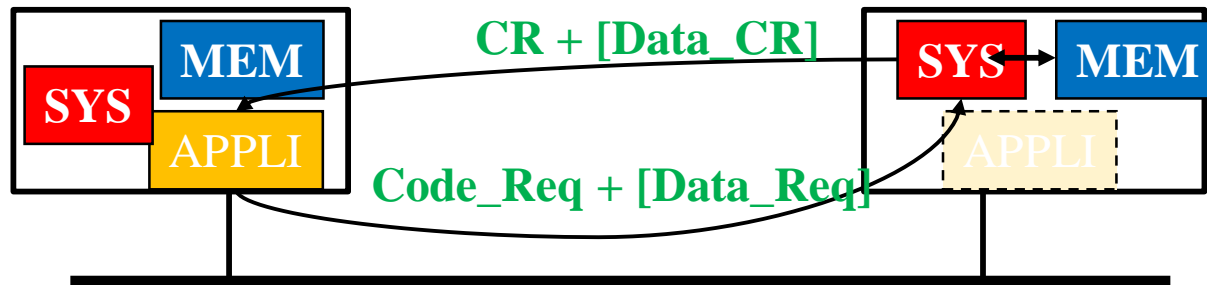
L'adresse réseau correspondant au coupleur réseau de plus basse adresse module (coupleur le plus à gauche dans le rack de la station) est considérée comme l'adresse principale de la station.





- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- **Exemples de codage de requête**
- **Encapsulation dans des paquets Modbus TCP/IP**

Messagerie industrielle UNITE (1) : client/serveur



- **Client**

- Actif
- Requête programmée

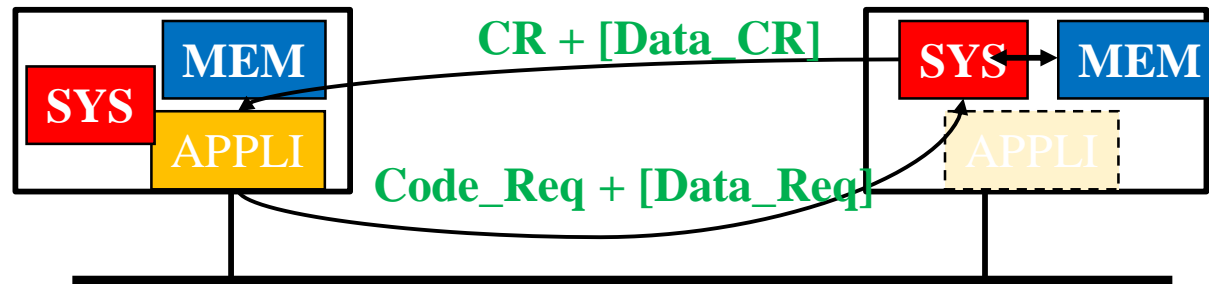
- **Serveur**

- Passif
- Il peut ne pas y avoir de programme
- Système d'exploitation qui met en oeuvre l'accès serveur

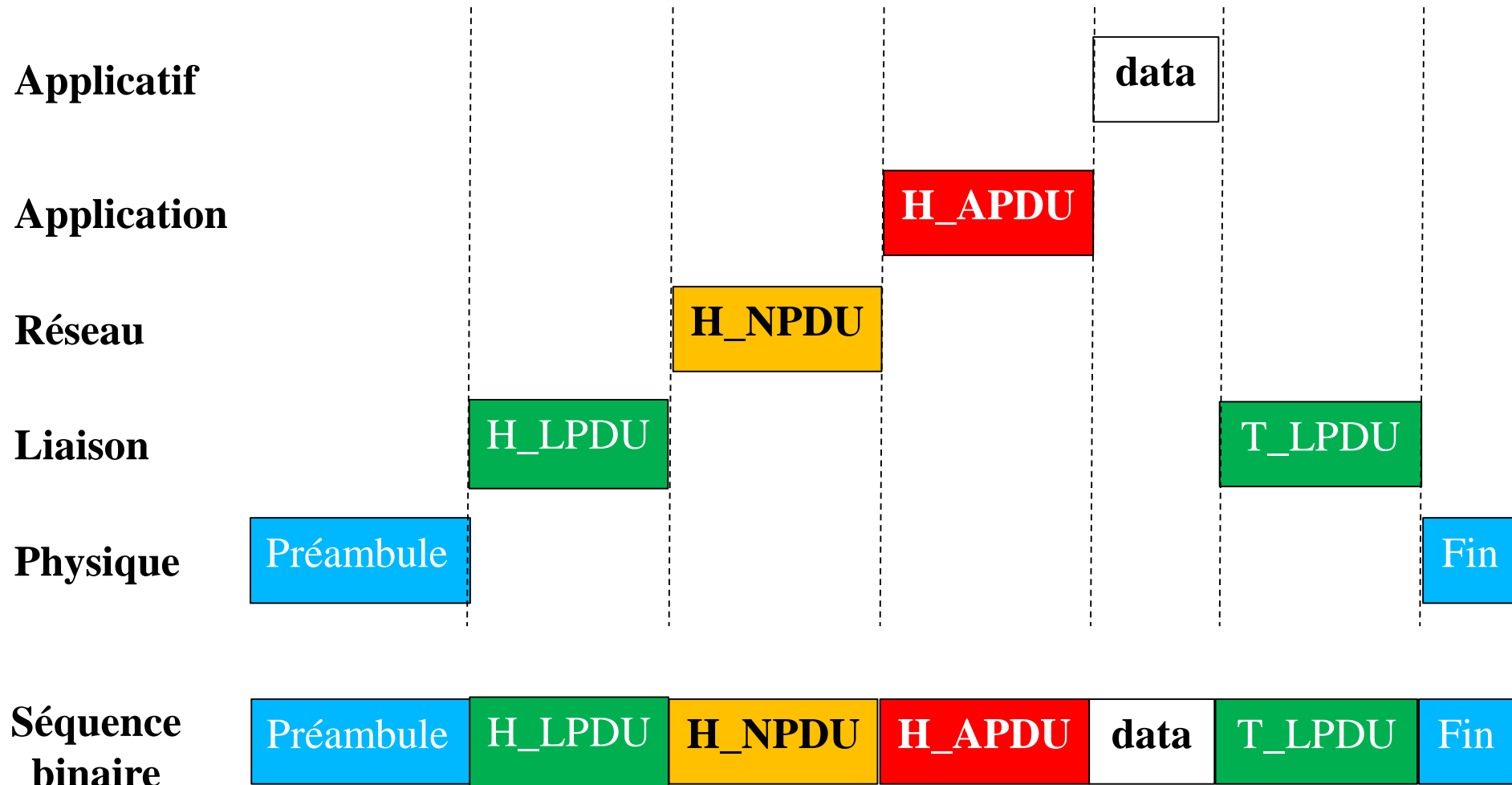
Nature des accès mémoire :

- Ecriture de variables
- Lecture de variables
- Téléchargement de programme

Messagerie industrielle UNITE (2) : client/serveur

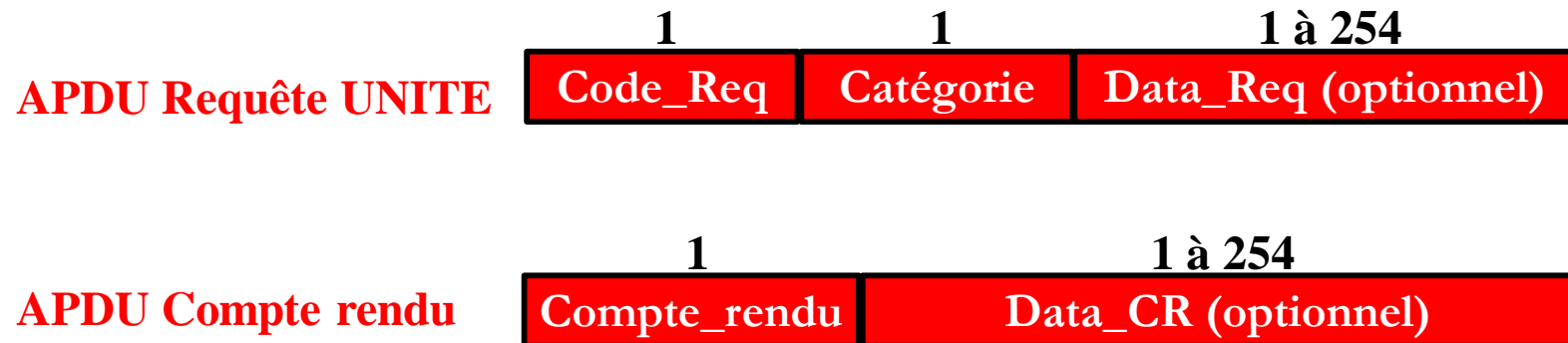


Compte-Rendu de la requête	Code requête + 30H Exception requête Miroir	L'opération a été exécutée par le serveur, des informations complémentaires précisent le résultat de l'opération
Compte-Rendu positif	FEh	L'opération s'est correctement déroulée et aucune information complémentaire n'est transmise dans la réponse.
Compte-Rendu négatif	FDH	L'opération n'a pu être exécutée par le serveur : requête inconnue, valeur hors norme, absence de configuration, ...



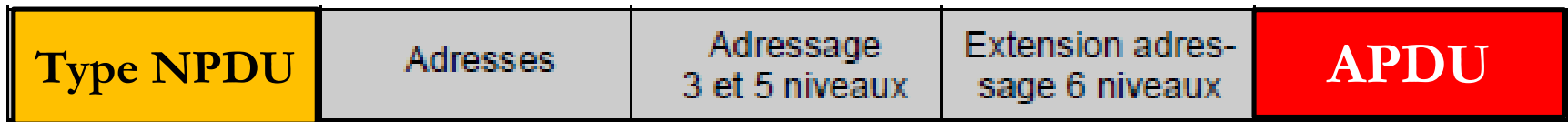
Messagerie industrielle UNITE (3) :

format APDU pour UNITE 1.1

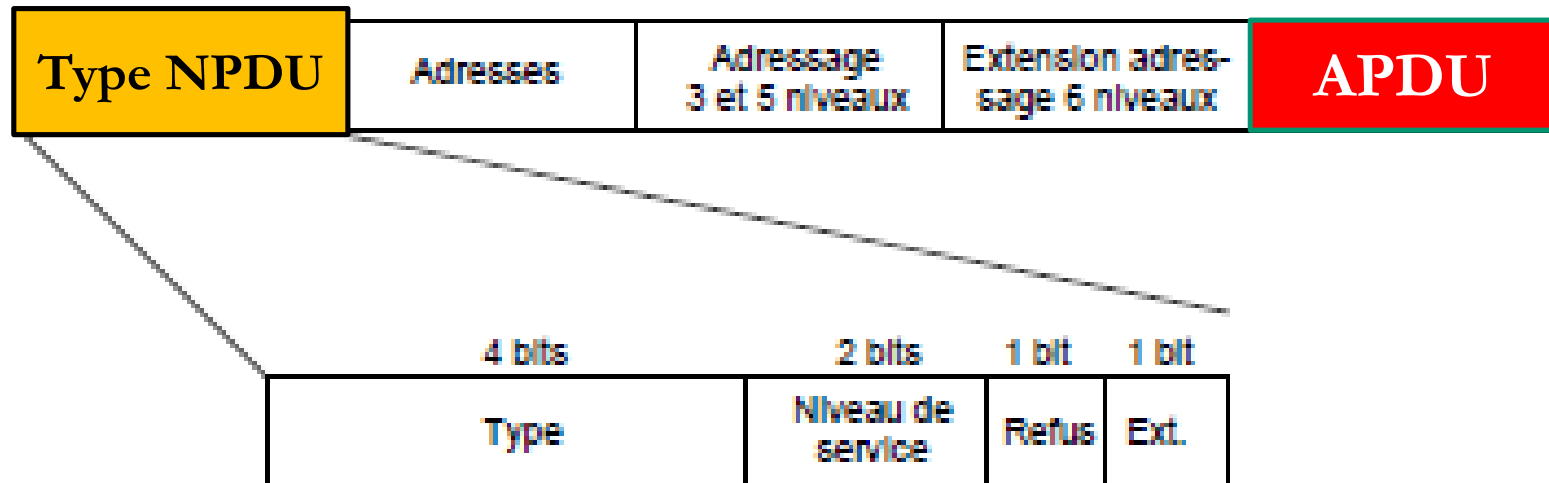


- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- **Exemples de codage de requête**
- **Encapsulation dans des paquets Modbus TCP/IP**

Couche réseau XWAY (1) : format NPDU



Couche réseau XWAY (2) : type NPDU

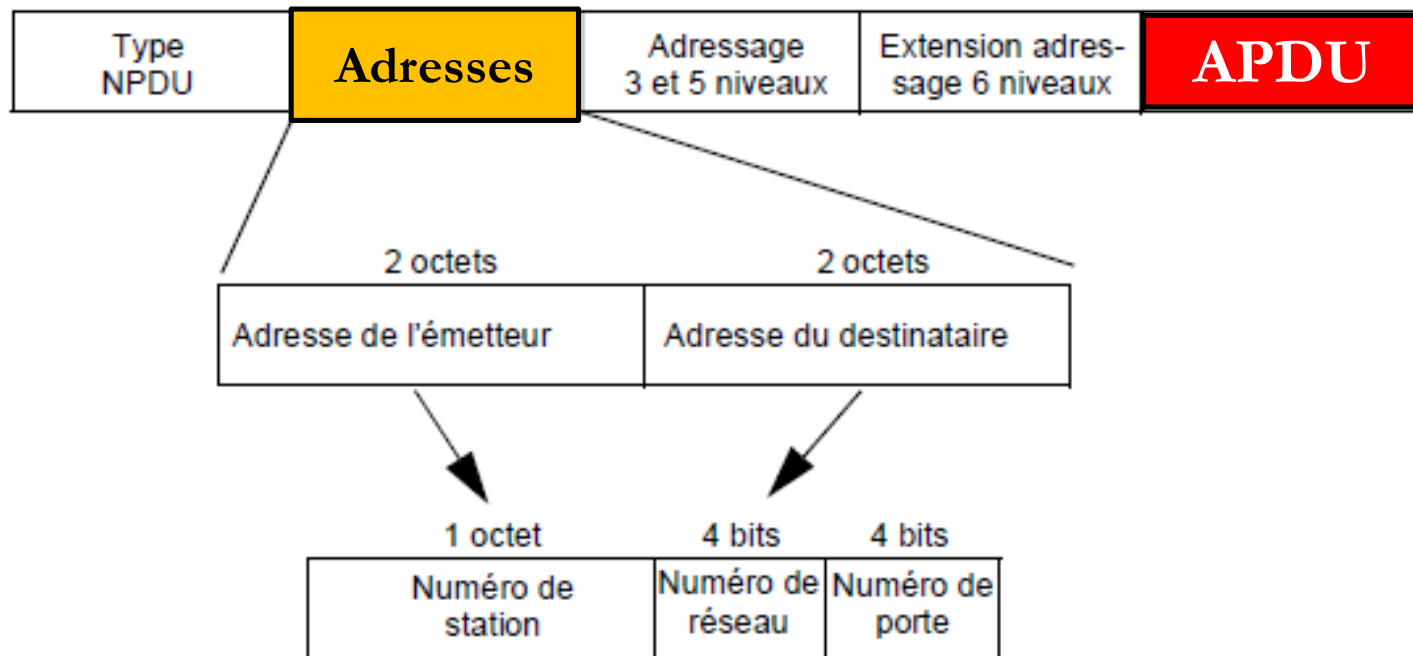


Couche réseau XWAY (3) :

Type NPDU

- **Type**
 - 0 à 14 : Réservé,
 - 15 : Données NPDU.
- **Niveau de service**
 - 0 : Standard,
 - 1 : Télégramme,
 - 2 à 3 : Réservé.
- **Refus**
 - 0 : Message accepté,
 - 1 : Message refusé.
- **Extension**
 - 0 : si l'un des équipements (émetteur ou destinataire) utilise l'adressage sur trois six niveaux pour l'échange concerné,
 - 1 : si l'un des équipements (émetteur ou destinataire) utilise l'adressage sur cinq niveaux pour l'échange concerné.
- **Exemples**
 - Type NPDU d'un message standard avec des adressages 3 niveaux émetteur et destinataire : 0xF0
 - Type NPDU d'un message standard avec adressage 5 ou 6 niveaux d'un équipement : 0xF1
 - Type NPDU d'un message refusé : 0xF2

Couche réseau XWAY (4) : adressage 3 niveaux

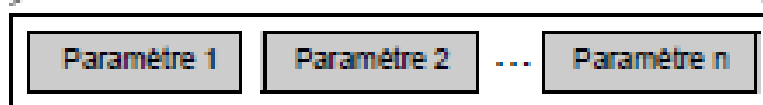


Couche réseau XWAY (5) :

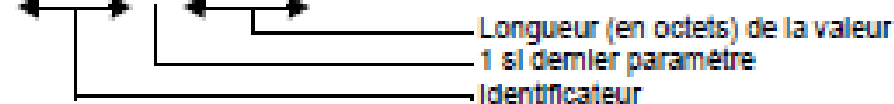
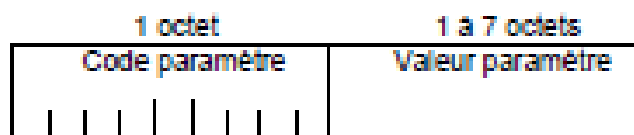
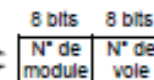
adressage 3 niveaux

- **Numéro de station**
 - Il correspond au numéro de station (de l'émetteur ou du destinataire).
 - Le numéro destination doit être compris entre 0 et 63 (la valeur 255 est réservée à la diffusion).
- **Numéro de réseau**
 - Correspond au numéro du réseau (de l'émetteur ou du destinataire).
 - Le numéro de réseau doit être compris entre 0 et 127.
 - Lorsque la valeur est supérieure à 15, une extension d'adresse est nécessaire.
- **Numéro de porte**
 - Correspond au numéro de porte visée dans l'entité destinataire de l'échange :
 - 0 accès au système de l'équipement (fonction SYS),
 - 1, 2, 3 accès à la prise terminal d'un automate programmable,
 - 5 accès à un module de communication (adressage hiérarchisé sur cinq niveaux),
 - 8 accès à un module de communication (adressage hiérarchisé sur six niveaux),
 - 11, 12, 13 accès au terminal connecté au point de raccordement 63 sur FIPIO,
 - 16 à 239 accès à l'application (bloc fonction texte, OF de communication, ...),
 - les autres valeurs sont réservées,
- **Remarque : Lorsque la valeur est supérieure à 15, une extension d'adresse est nécessaire.**

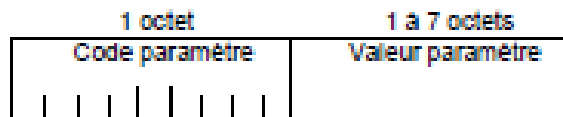
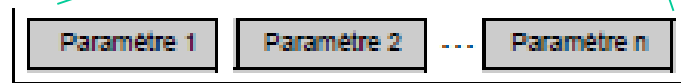
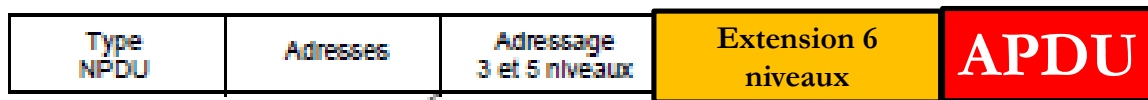
Couche réseau XWAY (6) : adressage 5 niveaux



Valeur paramètre	Identificateur	Longueur
N° porte émettrice si >15	0	1
N° porte destinataire si >15	1	1
N° réseau émetteur si >15	2	1
N° réseau destinataire si >15	3	1
N° module et n° voie de l'émetteur	4	2
N° module et n° voie du destinataire	5	2



Couche réseau XWAY (7) : adressage 6 niveaux



Longueur (en octets) de la valeur
1 si dernier paramètre
Identificateur

Valeur paramètre	Identificateur	Longueur	
Adresse émettrice physiquement compactée	6	2	4 bits Sélectionneur, 4 bits Référence, 8 bits Pt de raccord
Adresse destinataire physiquement compactée	7	2	
Adresse émettrice physiquement étendue	6	3	8 bits Sélectionneur, 8 bits Référence, 8 bits Pt de raccord
Adresse destinataire physiquement étendue	7	3	

- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- Exemples de codage de requête
- Encapsulation dans des paquets Modbus TCP/IP

Requête UNITE (1)

Rubrique	Nom de la requête	Code requête (héxa)	Code compte rendu (héxa)
Usage général	IDENTIFICATION	0F	3F
Usage général	READ_CPU	4F	7F
Usage général	PROTOCOL_VERSION	30	60
Usage général	MIRROR	FA	FB
Objets standards	READ_INTERNAL_BIT	0	30
Objets standards	WRITE_INTERNAL_BIT	10	FE
Objets standards	FORCE_INTERNAL_BIT	1B	FE
Objets standards	READ_INTERNAL_WORD	4	34
Objets standards	WRITE_INTERNAL_WORD	14	FE
Objets standards	READ_INTERNAL_DWORD	40	70
Objets standards	WRITE_INTERNAL_DWORD	46	FE
Objets standards	READ_CONSTANT_WORD	5	35
Objets standards	READ_CONSTANT_DWORD	41	71
Objets standards	READ_SYSTEM_BIT	1	31
Objets standards	WRITE_SYSTEM_BIT	11	FE
Objets standards	READ_SYSTEM_WORD	6	36
Objets standards	WRITE_SYSTEM_WORD	15	FE
Objets standards	READ_GRAFCET_BIT	2A	5A

Requête UNITE (2)

Rubrique	Nom de la requête	Code requête (héxa)	Code compte rendu (héxa)
Module d'E/S	READ_DIGITAL_MODULE_IMAGE	49	79
Module d'E/S	WRITE_DIGITAL_MODULE_IMAGE	4A	7A
Module d'E/S	READ_STATUS_MODULE	44	74
Module d'E/S	READ_IO_CHANNEL	43	73
Module d'E/S	WRITE_IO_CHANNEL	48	78
Objets génériques	READ_GENERIC_OBJECT	82	B2
Objets génériques	WRITE_GENERIC_OBJECT	83	B3
Objets génériques	READ_OBJECT	36	66
Objets génériques	WRITE_OBJECT	37	FE
Objets génériques	READ_OBJECT_LIST	38	68
Modes de marche	RUN	24	FE
Modes de marche	STOP	25	FE
Modes de marche	INIT	33	63

Requête UNITE (3)

Rubrique	Nom de la requête	Code requête (héxa)	Code compte rendu (héxa)
Transfert de données	OPEN_DOWNLOAD	3A	6A
Transfert de données	WRITE_DOWNLOAD	3B	6B
Transfert de données	CLOSE_DOWNLOAD	3C	6C
Transfert de données	OPEN_UPLOAD	3D	6D
Transfert de données	READ_UPLOAD	3E	6E
Transfert de données	CLOSE_UPLOAD	3F	6F
Transfert de données	BACKUP	45	75
Sémaphores	RESERVE	1D	FE
Sémaphores	RELEASE	1E	FE
Sémaphores	I_AM_ALIVE	2D	FE

Requête UNITE (4) : WRITE_OBJECT

Code requête 37h	Code catégorie 0 → 7	Segment de l'objet	Type objet	Instance du premier objet	Nombre d'objets à écrire	Données
1 octet	1 octet	1 octet	1 octet	1 mot	1 mot	Table de mots

Paramètres	Type	Commentaires
Code requête	octet	37h
Code catégorie	octet	0 à 7
Segment de l'objet	octet	voir <i>Objets accessibles</i> , p. 189.
Type objet	octet	voir <i>Objets accessibles</i> , p. 189.
Instance du premier objet	mot	contient l'adresse du premier objet à écrire.
Nombre d'objets à écrire	mot	nombre d'objets à écrire.
Données	Table de mots	données à écrire, le format dépend du type d'objet et du nombre d'objet à écrire.

Requête UNITE (5) : WRITE_OBJECT

Code
réponse
FEh

1 octet

Paramètres	Type	Commentaires
Code réponse négatif	mot	FDh <ul style="list-style-type: none">● automate non configuré,● segment inconnu du système,● type d'objet non présent dans le segment,● instance du dernier objet hors des limites de la configuration de PL7,● nombre d'objets à écrire égal à zéro.
Code réponse positif	mot	FEh

Requête UNITE (6) : Objets accessibles

Libellé objet	Segment	Type	Instance
Espace bit			
%Mi	64h	5	i : numéro de bit interne (nombre de bits multiple de 8).
%Si	64h	6	i : numéro de système interne (nombre de bits multiple de 8).
Espace données internes			
%MWi	68h	7	i : numéro du mot interne
%MDi	68h	8	i : numéro du double mot interne
Espace données constantes			
%KWi	69h	7	i : numéro du mot constant
%KDi	69h	8	i : numéro du double mot constant
Espace données système			
%SWi	6Ah	7	i : numéro du mot système
Espace système TSX7/TSX Premium/TSX Micro			
Horodateur	80h	1	03h : date et heure courante au format BCD,

Requête UNITE (7) :

Exemple de requête

- **37 07 68 07 0A 00 02 00 0B 00 0C 00**

Format	Code	Description
Segment de l'objet	68h	segment des objets à écrire (espace mots)
Type de l'objet	07h	type de l'objet (%MWi)
Instance du premier objet	0A 00h	instant du premier objet soit %MW10 (10=0Ah)
Nombre d'objet à écrire	02 00h	nombre d'objet à écrire, ici c'est 2
Données [Nombre d'objet]	0B 00h	valeur à écrire dans %MW10 vaudra 0B00h soit 11
Données [Nombre d'objet]	0C 00h	valeur à écrire dans %MW11 vaudra 0C00h soit 12

Requête UNITE (8) : READ_OBJECTS

Code requête 36h	Code catégorie 0 → 7	Segment de l'objet	Type objet	Instance du premier objet	Nombre d'objets à lire
1 octet	1 octet	1 octet	1 octet	1 mot	1 mot

Paramètres	Type	Commentaires
Code requête	octet	36h
Code catégorie	octet	0 à 7
Segment de l'objet	octet	voir <i>Objets accessibles</i> , p. 184
Type objet	octet	voir <i>Objets accessibles</i> , p. 184
Instance du premier objet	mot	Contient l'adresse du premier objet à lire.
Nombre d'objets à lire	mot	Nombre d'objets à lire.

Requête UNITE (9) : READ_OBJECTS

Code réponse 66h	Type de l'objet	Données
1 octet	1 octet	Table de mots

Paramètres	Type	Commentaires
Code réponse négatif	mot	FDh <ul style="list-style-type: none"> ● automate non configuré, ● segment inconnu du système, ● type d'objet non présent dans le segment, ● instance du dernier objet hors des limites de la configuration de PL7, ● nombre d'objets trop important pour le buffer de réponse. ● nombre d'objets à écrire égal à zéro, ● nombre de bits à lire non multiple de 8,
Code réponse positif	mot	66h
Type de l'objet	octet	Identique au champ de même nom dans la requête.
Données	Table de mots	valeur des objets lus (le format dépend du type d'objet et du nombre d'objets lus).

Requête UNITE (10) : READ_OBJECTS

Libellé objet	Segment	Type	Instance
Espace bit			
%Mi	64h	5	i : numéro de bit interne (nombre de bits multiple de 8).
%Si	64h	6	i : numéro de système interne (nombre de bits multiple de 8).
Espace données internes			
%MWi	68h	7	i : numéro du mot interne
%MDi	68h	8	i : numéro du double mot interne
Espace données constantes			
%KWi	69h	7	i : numéro du mot constant
%KDi	69h	8	i : numéro du double mot constant
Espace données système			
%SWi	6Ah	7	i : numéro du mot système
Espace système TSX7/TSX Premium/TSX Micro			
Horodateur	80h	1	<ul style="list-style-type: none"> ● 03h : date et heure courante au format BCD, ● 04h : date et heure du dernier arrêt au format BCD,

Requête UNITE (11) : Exemple de requête

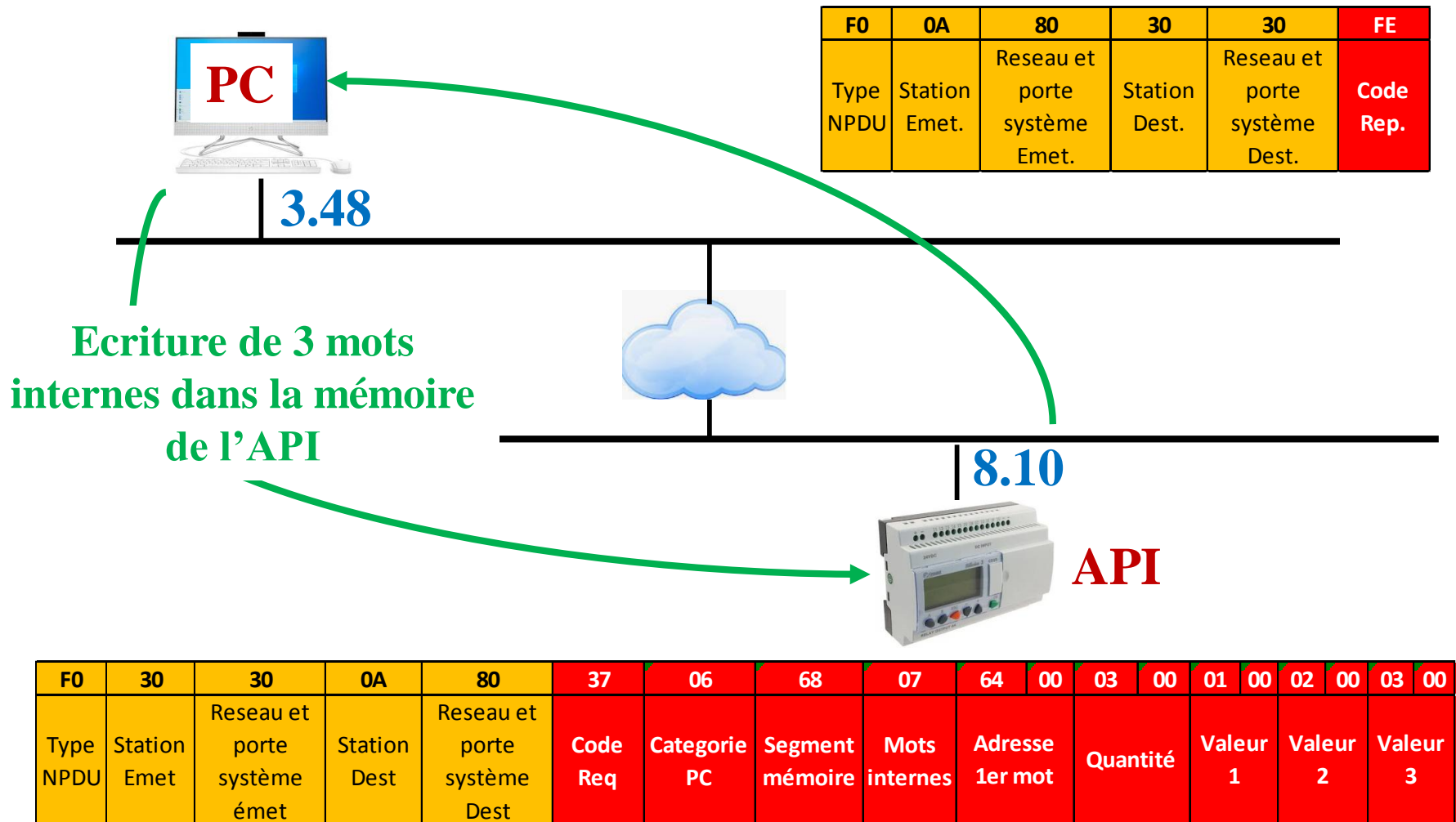
- **36 07** 80 01 03 00 01 00 00 04 14 47 10 19 10 01 20

Format	Code	Description
Segment de l'objet	80h	segment de l'objet
Type de l'objet	01h	type de l'objet
Instance du premier objet	03 00h	Attention c'est au Format Intel, donc on lit 00 03h qui correspond à la date et l'heure courante au format BCD
Nombre d'objets à lire	01 00h	Attention c'est au Format Intel, donc on lit 00 01h qui correspond au nombre d'objets à lire
Données[nombre d'objets]	00 04 14 47 10 19 10 01 20h	<p>correspond aux 9 octets de l'horodateur:</p> <ul style="list-style-type: none"> ● 00 : dixième de seconde. ● 04 : vendredi, ● 14 : secondes, ● 47 : minutes, ● 10 : heure, ● 19 : le 19 octobre, ● 10 : mois d'octobre, ● 01 20h : année 2001,

- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- **Exemples de codage de requête**
- **Encapsulation dans des paquets Modbus TCP/IP**

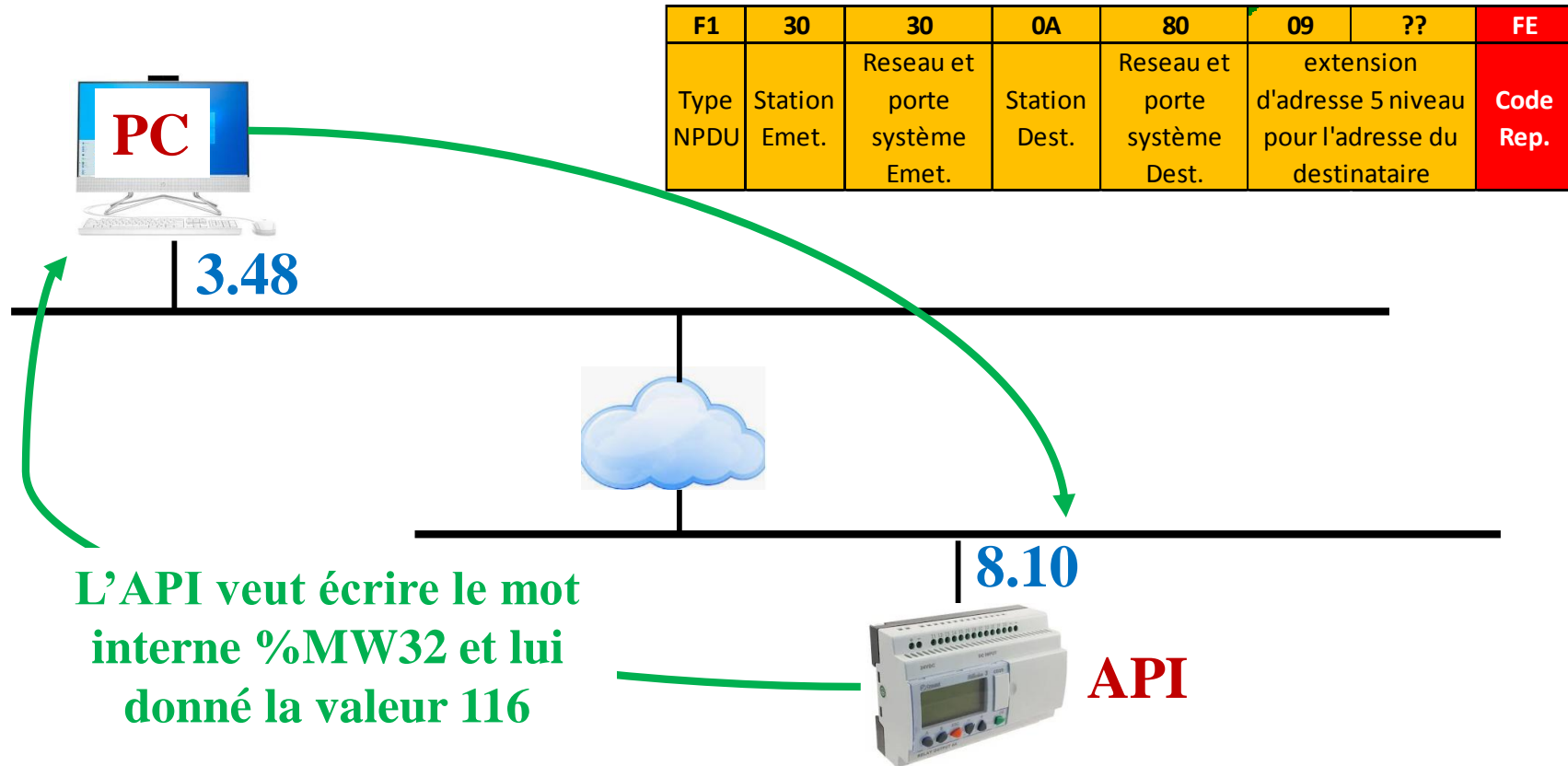
Exemples de codage de requête

UNITE (1) : cas simple d'un PC



Exemples de codage de requête

UNITE (2) : cas d'un API

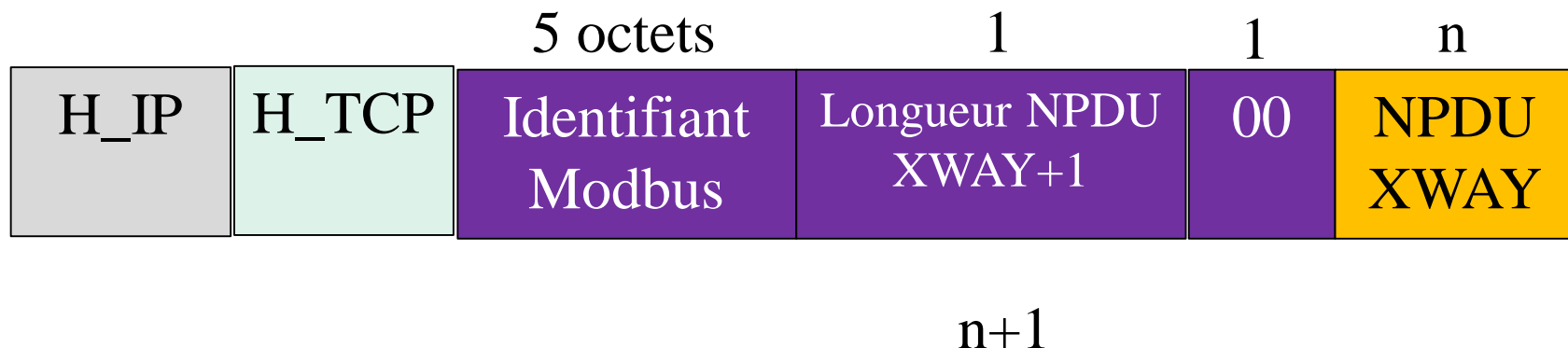


F1	0A	80	30	30	1E	??	37	07	68	07	20	00	01	00	74	00
Type NPDU	Station Emet.	Reseau et porte système Emet.	Station Dest.	Reseau et porte système Dest.	extension d'adresse 5 niveau pour l'adresse de l'émetteur		Code Req	Categorie API	Segment mémoire	Mots internes	Adresse 1er mot		Quantité		Valeur 1	

- **Présentation du fonctionnement des automates programmables industriels**
- **Architecture XWAY**
- **Adressage XWAY**
- **Messagerie Industrielle UNITE**
- **Couche réseau XWAY**
- **Requête UNITE**
- **Exemples de codage de requête**
- **Encapsulation dans des paquets Modbus TCP/IP**

Encapsulation des paquets XWAY dans Modbus TCP/IP

- Les modules ETHWAY des API Schneider embarquent un serveur TCP qui est à l'écoute des connexions sur le port TCP 502 (port modbus TCP/IP)
- Utilisation de Modbus TCP/IP pour faire du tunneling pour réseau XWAY
- Identifiant ModbusTCP/IP : *00 00 00 01 00*



FIN