R120 – Structure des API

exemple de l'automate S7-1500

Licence Pro Rob&IA

Laurent ROY

1 Introduction

Gamme Simatic

- **S7 1500 :** automate modulaire
- **S7 1200** : modèle compact
- **Logo :** modèle compact orienté tertiaire

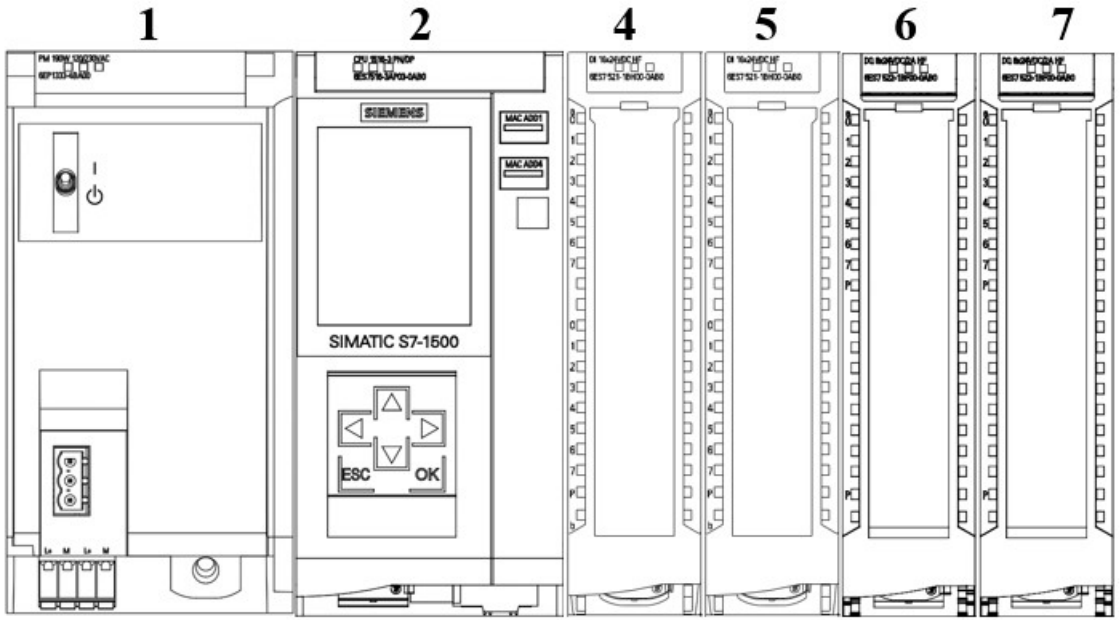
} se programment avec le logiciel **TIA Portal** de Siemens.

Composition d'un S7 1500 :

- modules **PS** (alimentation),
- **CPU** (unité centrale),
- **SM** (module de signaux d'entrées/sorties),
- modules **FM** pour les fonctions spéciales (cmde de moteur pas à pas),
- processeurs de communication **CP** pour les liaisons réseau.

1 - Présentation de l'automate S7-1500

N° d'emplacement →



Type de module →

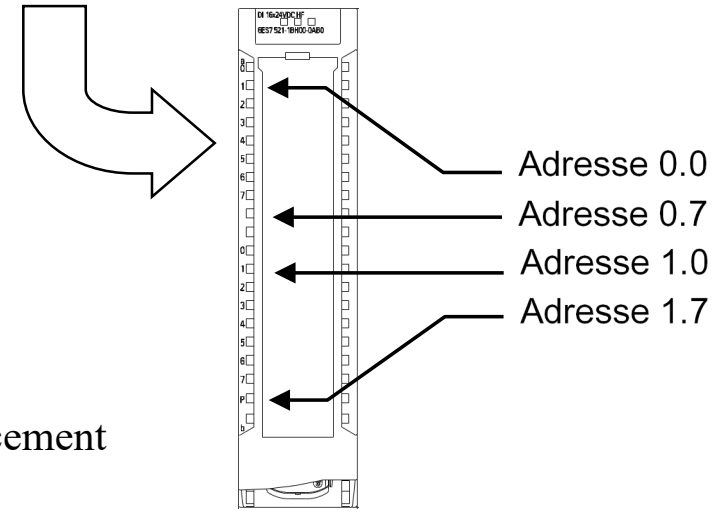
Adresses →

PS	CPU	SM	SM	SM	SM
		0.0 à 3.7	4.0 à 7.7	8.0 à 11.7	12.0 à 15.7

1 - Présentation de l'automate S7-1500

Une entrée ou une sortie est désignée dans le programme à l'aide d'une adresse qui indique clairement quel est son emplacement sur l'automate. Cette adresse est composée :

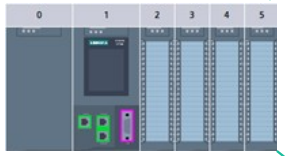
- d'un **%** indiquant une adresse automate,
- d'une lettre indiquant la nature de la variable : **I** pour une entrée et **Q** pour une sortie
- d'un chiffre, appelé **adresse d'octet**, qui indique l'emplacement du module (de 0 à 31),
- d'un point (**.**)
- d'un chiffre, appelé **adresse de bit**, qui indique l'emplacement de la variable sur le module (de 0 à 7).



Nota : _ L'emplacement 3 est réservé au coupleur **IM** pour une configuration multichâssis
_ 4 adresses d'octet sont réservés à chaque emplacement. En cas d'utilisation de modules d'entrées/sorties à 16 voies, on perd 2 adresses d'octet par emplacement.

1 - Présentation de l'automate S7-1500

Configuration de l'automate : **Hardware Config** , il faut inclure les différents modules et les paramétrer.



Module	...	Empla..	Adresse I	Adresse Q	Type	N° d'article	Firmware
PM 190W 120/230VAC	0	0			PM 190W 120/230VAC	6EP1333-4BA00	
▼ PLC_1	0	1			CPU 1516-3 PN/DP	6ES7 516-3AN02-0AB0	V2.8
▶ Interface PROFINET_1	0	1 X1			Interface PROFINET		
▶ Interface PROFINET_2	0	1 X2			Interface PROFINET		
DI 32x24VDC HF_1	0	2	0...3		DI 32x24VDC HF	6ES7 521-1BL00-0AB0	V2.2
DQ 32x24VDC/0.5A HF_1	0	3		0...3	DQ 32x24VDC/0.5A HF	6ES7 522-1BL01-0AB0	V1.1
AI 8xU/I/RTD/TC ST_1	0	4	4...19		AI 8xU/I/RTD/TC ST	6ES7 531-7KF00-0AB0	V2.2
AQ 4xU/I ST_1	0	5		4...11	AQ 4xU/I ST	6ES7 532-5HD00-0AB0	V2.2

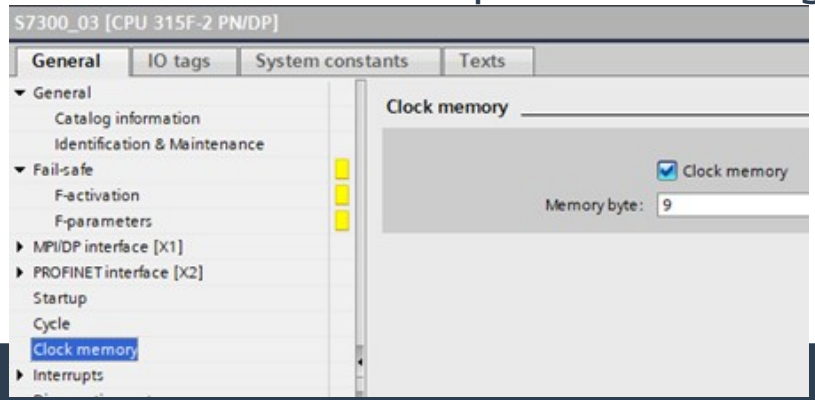
1 - Présentation de l'automate S7-1500

Configuration de l'automate : **Hardware Config** , il faut inclure les différents modules et les paramétrer. Exemples de paramétrages :

_ module entrée analogique : définir le type de transmetteur, et la plage de variation du signal.

_ module CPU : Il faut définir la durée du cycle de surveillance (chien de garde), on peut également définir le mémento de cadence et les zones de rémanence.

- Le **mémento de cadence** est un octet de mémoire dont chaque bit va changer périodiquement. Si %MB9 est choisi, alors :
  - %MB9.7 fréquence de 0,5 Hz.
  - %MB9.6 fréquence de 1 Hz.
  - ...



2 - Types de données utilisées

Données	Description	Nbre de bits	Etendue
BOOL	Booléen	1	0 ou 1
BYTE	Octets de 8 bits	8	Pas d'étendue numérique pour ce type de données
WORD	Mot de 16 bits	16	
DWORD	Mot double de 32 bits	32	
INT	Entier signé	16	-32 768 à + 32767
DINT	Double entier signé	32	**-2 147 483 648 à 2 147 483 647**
REAL	Nombre réel à virgule flottante	32	**-10^{-38} à 10^{38}**

2 - Types de données utilisées

Pour visualiser les valeurs de taille octet, mot ou double mot, on peut utiliser différents formats:

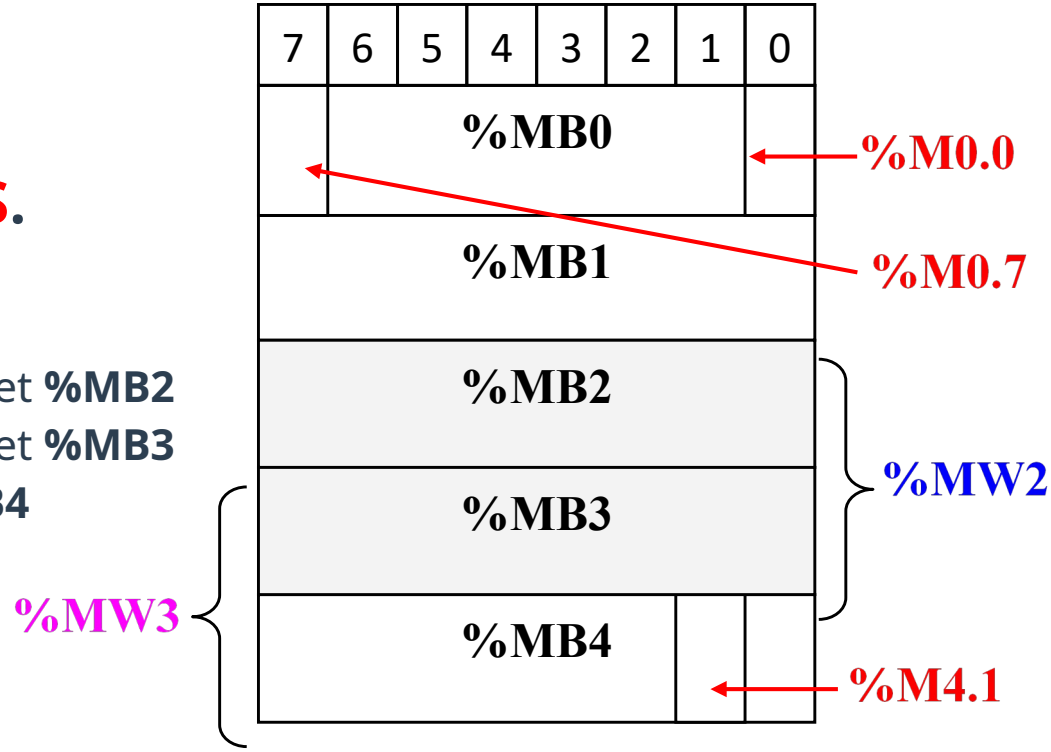
format décimal: **20047**
format binaire: **2#0100 1110 0100 1111**
format hexadécimal: **16#4E4F**
format réel ou virgule flottante: **+2.0047E4**
format ASCII: **'vingt mille quarante sept'.**

3 - Adressages des zones de mémoire de la CPU

La mémoire du **S7 1500** est organisée
à partir d'**OCTETS** ou de **BYTE**.

ATTENTION AUX RECOUVREMENTS.

Exemples :
%MW2 est un mot de 16 bits composé de **%MB3** et **%MB2**
%MW3 est un mot de 16 bits composé de **%MB4** et **%MB3**
%M4.1 est le deuxième bit de l'octet ou byte **%MB4**
%M1.8 n'existe pas !!!.



3 - Adressages des zones de mémoire de la CPU

1.1 Accès à un bit

Exemple: ***%I 1 . 3***

- **%I**: identificateur de zone (**Mémoire Image des Entrées**),
- **adresse d'octet**: octet 1 (2ème octet de la MIE)
- **rang du bit sur l'octet**: bit 3 de l'octet (de 0 à 7).

%IB1	7	6	5	4	**3**	2	1	0

1.2 Accès à un octet de 8 bits :

Exemple : ***%M B 81***

- **%M**: identificateur de zone (mémoire variables internes)
- **B** accès à un octet (**Byte**)
- **81** : adresse d'octet.

octet %M B 81:	**%M 81.7**	**%M 81.6**	**%M 81.5**	**%M 81.4**	**%M 81.3**	**%M 81.2**	**%M 81.1**	**%M 81.0**

3 - Adressages des zones de mémoire de la CPU

1.1 Accès à un mot de 16 bits :

Exemple ***%M W 2*** *ATTENTION toujours prendre des adresses PAIRES (recouvrement)*

- **%M**: identificateur de zone (mémoire variables internes)
- **W** accès à un mot (Word)
- **2 :** adresse de l'octet de poids fort du mot.

mot %MW2	%MB 2 (octet de poids fort)								%MB 3 (octet de poids faible)							
	M2.7	M2.6	M2.5	M2.4	M2.3	M2.2	M2.1	M2.0	M3.7	M3.6	M3.5	M3.4	M3.3	M3.2	M3.1	M

1.2 accès à un double mot de 32 bits :

Exemple ***%M D 108*** *ATTENTION prendre des adresses multiples de quatre (recouvrement)*

- **%M**: identificateur de zone (mémoire variables internes)
- **D** accès à un double mot (Double word)
- **108** : adresse de l'octet de poids fort du double mot.

%M D 108	%M B 108	%M B 109	%M B 110	%M B 111

4 Table d'adressage

1		S1_BoxLong	Bool	%I0.0
2		S2_BoxLow	Bool	%I0.1
3		S3_BoxHigh	Bool	%I0.2
4		S4_BoxOut	Bool	%I0.3
5		Emergency_stop	Bool	%I0.4
6		Sel_Auto	Bool	%I0.5
7		Convoyer1	Bool	%Q0.0
8		Convoyer2	Bool	%Q0.1
9		Stop_Blade	Bool	%Q0.2
10		Stop_light	Bool	%Q0.3

La table d'adressage permet d'associer des noms de variables, appelés « **mnémoniques** » ou ***tags*** à des données :

- nom (*tag)*,
- type (BOOL, BYTE, WORD, DWORD, INT, DINT, REAL …)
- adresse (%I*i.j*, %Q*i.j*, %MB*i* , %MDi, %IW*i*,…)
- Si la donnée est **rémanente** ou non. Une donnée rémanente conservera sa valeur après une coupure d'alimentation de l'automate,
- Les accès qui lui sont associés. Une variable peut être accessible en lecture/écriture depuis l'extérieur de l'automate, ou n'être pas visible.
- Un commentaire, facultatif mais utile, qui décrit le rôle de la variable dans le système piloté.

5 Programmation des S7-1500

3 types de blocs fondamentaux :

- le **bloc programme principal** appelé **OB1** (Bloc d'Organisation n°1). exécuté de **manière cyclique** par la CPU de l'automate. **OB1** est toujours scruté. Il doit aussi gérer l'appel des sous programmes.

- les **blocs sous programmes** (appelés **FC** ou **FB**). Ils sont exécutés uniquement lorsque le programme principal les appelle -> permettent de structurer l'application.

- les **programmes d'interruptions** (autres **OB** que **OB1**). Ils sont exécutés lorsque se produit l'événement d'interruption correspondant à ce programme. Par exemple :
  - **_OB 100** est un bloc qui est exécuté lorsque l'automate passe en RUN (démarrage à chaud).
  - **_OB 35** est un bloc qui peut être exécuté toutes les 100ms.
  - **_OB 80** est un bloc qui est exécuté en cas de débordement du chien de garde.

5 Programmation des S7-1500

- **<u>Les langages de programmation CEI 61131-3 du S7-1500 (rappels)</u>**

- Ce logiciel **TIA Portal** permet de programmer l'automate S7-1500 avec un des 5 langages normalisés (norme européenne **CEI 61131-3**) :
  - le langage à contacts nommé LADDER,
  - le langage liste d'instructions Instruction Liste IL,
  - le langage logigramme ou Function Block Diagram FBD,
  - Le langage Structured Text ST,
  - le langage Sequential Function Chart SFC.

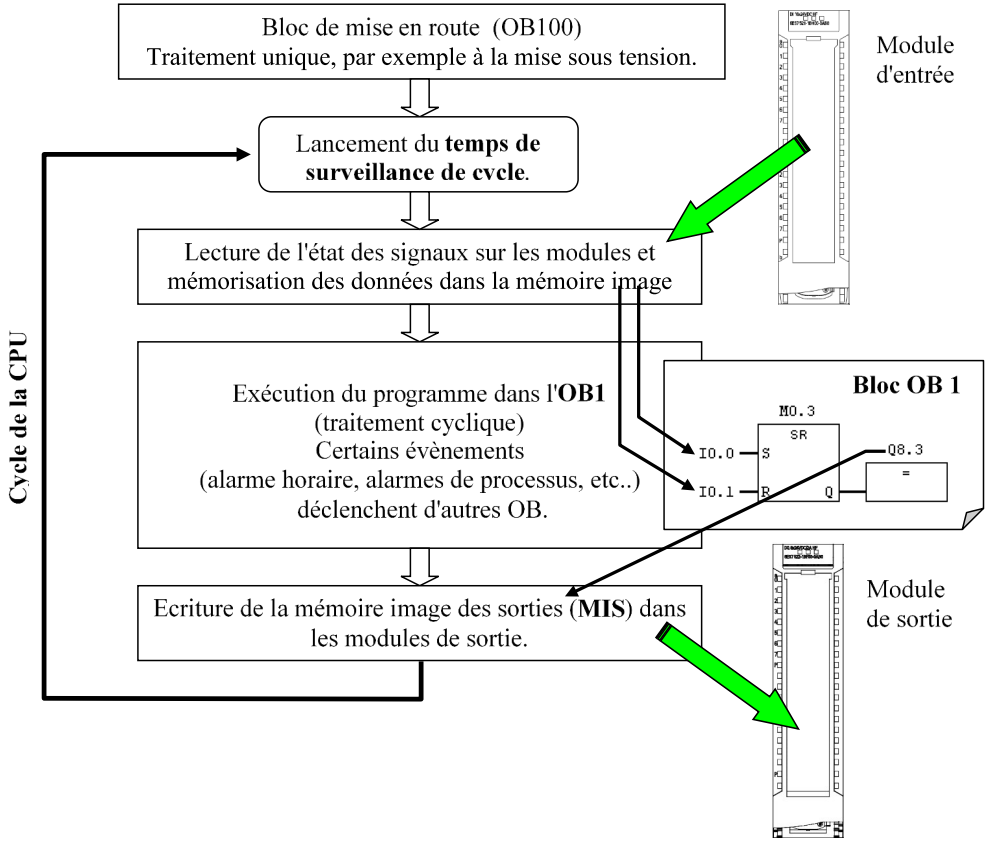
5 Programmation des S7-1500

Famille	Exemples
opérations combinatoires sur bits:	blocs ***NOT*** complément bit à bit d'un octet ou mot. blocs ***AND*** ou ***OR*** ET ou OU bit à bit d'un octet ou mot.
opérations de comparaison.	Blocs ***CMP ==*** égalité entre 2 mots. Blocs ***CMP >=*** supériorité entre 2 mots ; **<** infériorité entre 2 mots
opérations de temporisation.	Blocs ***TON, TOF, TP***
opérations de comptage.	Blocs ***CTU*** compteur incrémental (+1 à chaque front montant). Blocs ***CTD*** compteur décrémental (-1 à chaque front montant).

5 Programmation des S7-1500

Famille	Exemples
opérations arithmétiques.	Blocs **ADD, SUB, MUL, DIV** …
opérations de transfert	Bloc **MOVE** (opérateur d'affectation, permet également de changer de type de données)
opérations de décalage et de rotation.	Blocs de décalage à droite **SHR** (*Shift Right*) ; à gauche **SHR** (*Shift Left*) Blocs de rotation à droite **ROR** (*Rotate Right*) ; à gauche **ROL**
opérations de conversion.	Bloc **NORM_X** (convertir une valeur numérique en %) Bloc **SCALE_X** convertir un pourcentage une valeur numérique)

5.4 Traitement cyclique du programme



A la mise sous tension ou à la mise en marche (RUN), la CPU procède à une initialisation complète (démarrage avec l'OB100). le système d'exploitation efface les mémentos, les temporisations et les compteurs non rémanents, les alarmes de processus et de diagnostic mémorisées et il lance le temps de surveillance du cycle.

Cycle de scrutation

Le fonctionnement cyclique de la CPU comprend 3 étapes principales :

La CPU interroge l'état des signaux des modules d'entrées et actualise la mémoire image des entrées **MIE**.

Elle exécute le programme utilisateur avec ses différentes opérations.

Elle copie les valeurs de la mémoire image des sorties **MIS** dans les modules de sortie.

Fin

Bibliographie :
Boujat G., Anaya P. : Automatique industrielle en 20 fiches, Dunod
Documentation constructeur Siemens :
https://mall.industry.siemens.com