

SEMAINE 5

INITIATION AUX AUTOMATES SIEMENS

FICHE 45 : ADRESSAGE DES VARIABLES



Automation & Sense

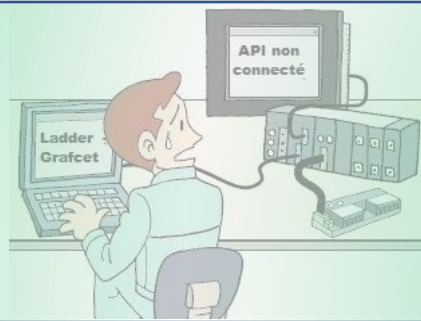


Objectifs :



Dans cette fiche, nous allons voir les différents types de variables qui existent sur TIA Portal et comment les adresser.

Nous verrons par la même occasion les différentes zones mémoires composants un automate Siemens.



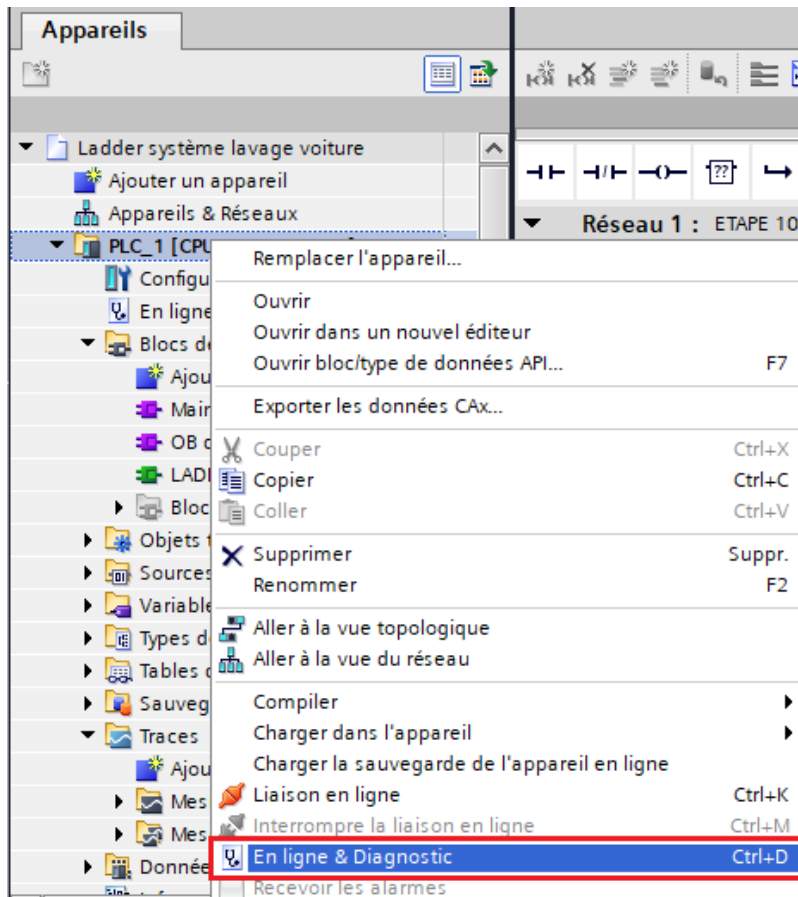
I) Les différents types de mémoires

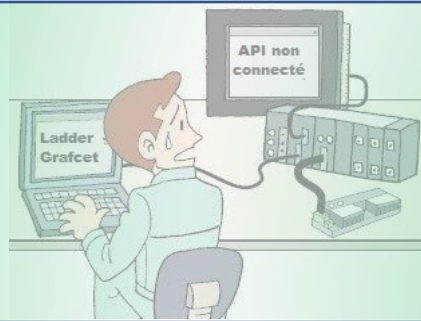
Les CPU Siemens comportent différents types de mémoires qui permettent de stocker le programme utilisateur et différents types de données dont des données de configuration. On distingue principalement 3 types de mémoires :

- **La mémoire de chargement** : c'est une mémoire non volatile utilisée pour stocker le programme utilisateur et les données de configuration. En effet, quand on transfère un programme sur un CPU, celui-ci est stocké dans un premier temps au niveau de la mémoire de chargement. La mémoire de chargement se présente souvent sous la forme d'une carte mémoire MMC que l'on insère au niveau du CPU.
- **La mémoire de travail ou RAM** : c'est une mémoire volatile permettant d'exécuter le programme de l'automate. Les données que contient cette mémoire sont perdues après une coupure de courant.
- **La mémoire rémanente** : C'est une mémoire non volatile permettant de stocker quelques données de la mémoire de travail. C'est l'utilisateur qui va spécifier au niveau du logiciel les données de la mémoire de travail à enregistrer dans la mémoire rémanente. Cela permet ainsi de ne pas perdre ces données en cas de coupure de courant.

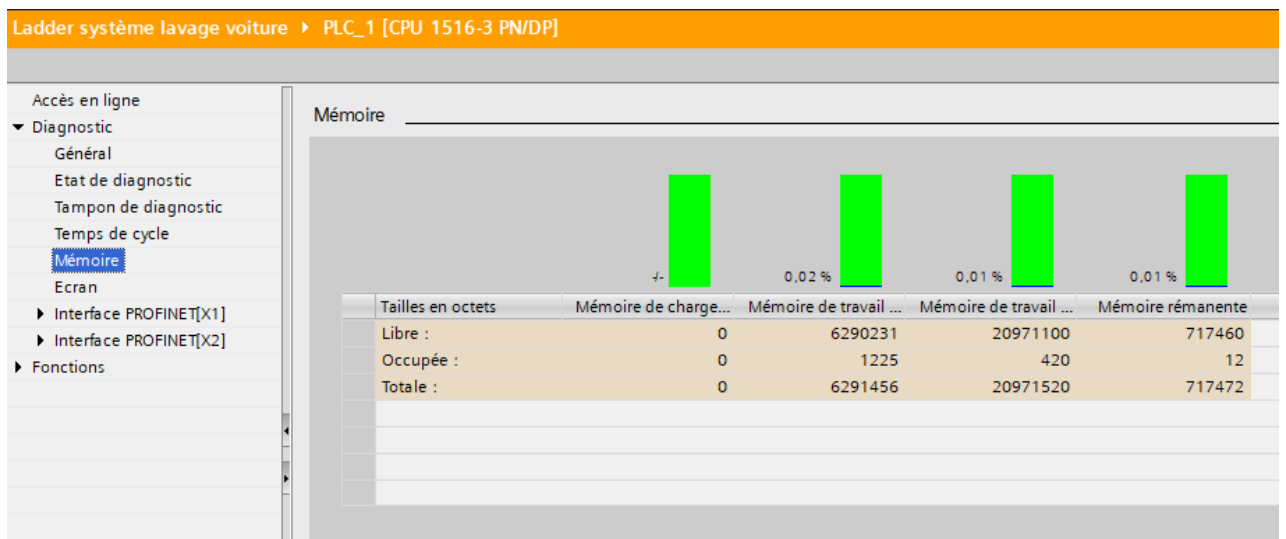


Pour visualiser l'utilisation de la mémoire d'un CPU sur TIA Portal, faites un clic droit sur le CPU et choisissez « **En ligne & diagnostic** »





Après avoir établi la liaison en ligne avec l'automate via le bouton « **Liaison en Ligne** », vous pourrez observer l'utilisation de la mémoire de l'automate comme montré sur l'image ci-dessous.

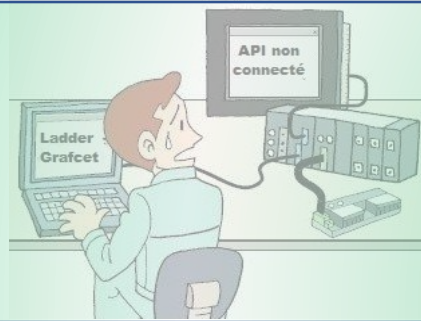


II) Les emplacements mémoires d'un automate

Durant l'exécution d'un programme automate, les données sont enregistrées dans différents emplacements mémoires :

- Input (I)
- Output (Q)
- Memento (M)
- Bloc de données (DB)
- Etc..

Le programme utilisateur pourra accéder à tous ces emplacements mémoires pour effectuer des opérations de lecture ou d'écriture.



Vous pouvez voir ci-dessous une liste de variables localisées dans plusieurs emplacements mémoires.

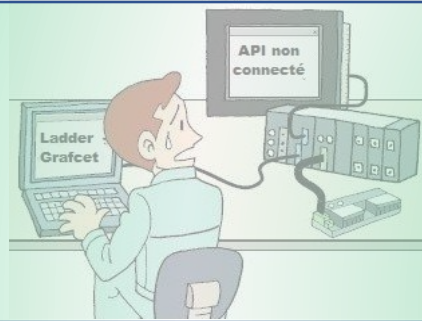
| Variables API | | | | | | | | |
|---------------|------|-----------------|---------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---|
| | Nom | Type de données | Adresse | Réma... | Acces... | Ecritu... | Visibl... | Commentaire |
| 12 | CNT2 | Bool | %M0.4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 13 | KM3 | Bool | %Q0.3 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | contacteur rotation des 3 rouleaux |
| 14 | KM1 | Bool | %Q0.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | contacteur avance portique |
| 15 | VCL | Bool | %Q0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Voyant vert |
| 16 | CL | Bool | %I1.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | capteur sélection catégorie véhicule |
| 17 | Cp | Bool | %I0.7 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | présence véhicule au poste de lavage |
| 18 | Ch | Bool | %I0.6 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | fin de course rouleau horizontal en haut |
| 19 | Cb2 | Bool | %I0.5 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | fin de course rouleau horizontal position basse |
| 20 | Cb1 | Bool | %I0.4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | fin de course rouleau horizontal mi-course bass |
| 21 | Cv | Bool | %I0.3 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | fin de course portique en avant |
| 22 | Cr | Bool | %I0.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | fin de course portique en arrière |
| 23 | S1 | Bool | %I0.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | BP départ cycle |
| 24 | KM2 | Bool | %Q0.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | contacteur recul portique |

III) Adressage d'une variable d'un automate

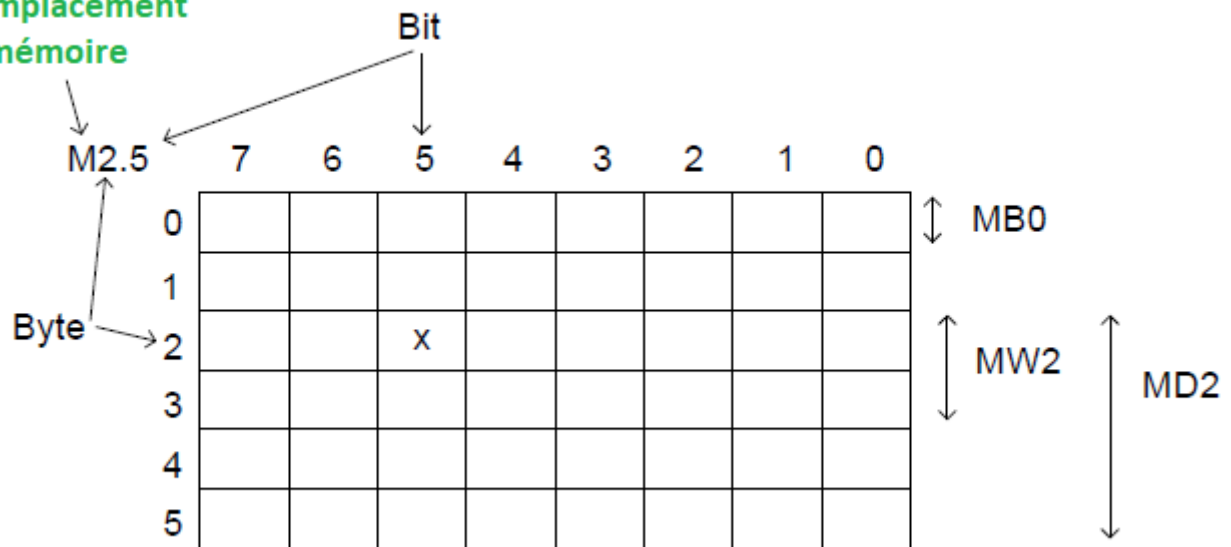
Au niveau du programme, les variables automates sont adressées d'une manière très spéciale. Il faut savoir que l'on distingue deux types d'adresses : les adresses symboliques et les adresses absolues. L'adressage symbolique permet d'adresser une variable juste avec un nom symbolique. Exemple : on pourra avoir **KM3** pour désigner le contacteur d'un moteur.

Une variable est adressée en fonction de son emplacement mémoire et du nombre de bits utilisés pour le coder. Une variable peut être codée sous un format « Byte », « Word » ou « Double Word ».

Si la variable est codée sur 8 bits on utilisera le format **B**, si la variable est codée sous 16 bits, on utilisera le format **W** et si la variable est codée sur 32 bits ou 4 octets on utilisera le format **D**.



Identifiant de l'emplacement mémoire

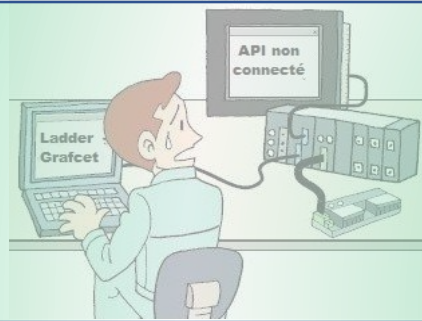


En désignant **M2.5**, on adresse la variable située au niveau du bit 5 de l'octet 2 du momento M. Si par exemple on avait désigné MB2, on aurait désigné tous les 8 bits de l'octet 2 du momento M.

IV) Etude de cas : le CPU S7-1214C

Le S7-1214C est un CPU de type compact comportant :

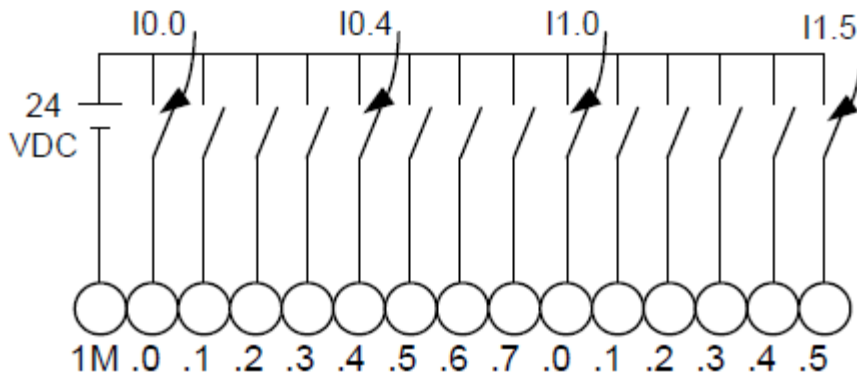
- 14 entrées digitales
- 10 sorties digitales
- 2 entrées analogiques

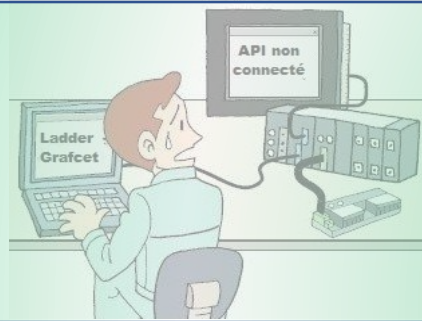


Les différents types de données disponibles au niveau de ce CPU sont les suivants :

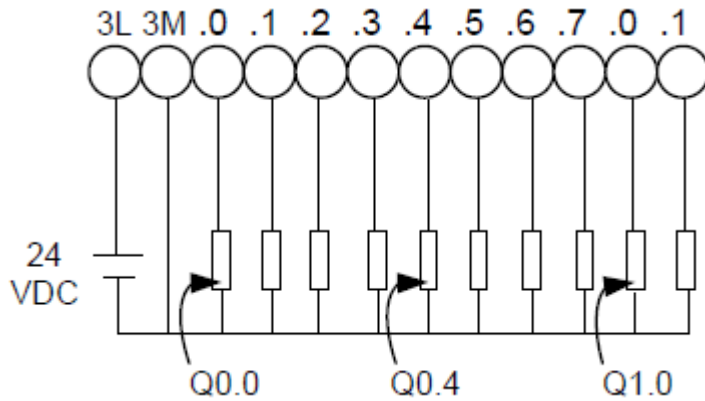
| Data type | Size |
|---------------------------------|---|
| Bool | 1 bit |
| Byte | 8 bits |
| Word | 16 bits |
| DWord | 32 bits |
| Char | 8 bits |
| Sint (short integer) | 8 bits (-128 to 127) |
| USInt (unsigned short integer) | 8 bits (0 to 255) |
| Int (integer) | 16 bits (-32768 to 32767) |
| UInt (unsigned integer) | 16 bits (0 to 65535) |
| Dint (double integer) | 32 bit (-2,147,483,648 to 2,147,483,647) |
| UDInt (unsigned double integer) | 32 bit (0 to 4,294,967,295) |
| Real (real) | 32 bit (+/- 1.18 x 10 ⁻³⁸ to +/- 3.40x 10 ³⁸) |
| LReal (long real) | 64 bits (+/- 2.23 x 10 ⁻³⁰⁸ to +/- 1.79x 10 ³⁰⁸) |
| Time (time) | Defined later |
| String (character string) | Variable |
| DTL (data and time long) | Defined later |

Les 14 entrées digitales du CPU S7-1214C sont adressées et câblées de la manière suivante :

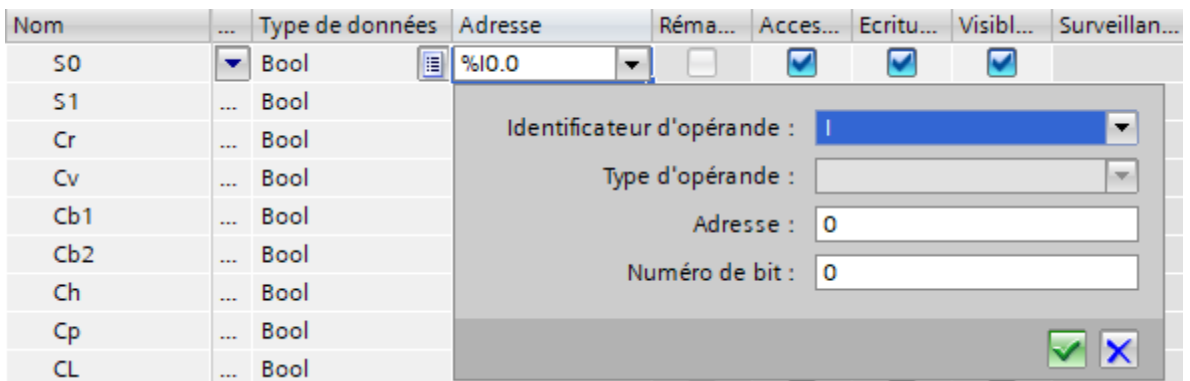


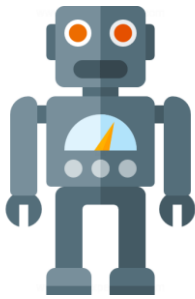


Les 10 sorties digitales sont adressées et câblées de cette manière :



Sur TIA Portal, pour adresser la première entrée de l'automate, on pourra utiliser comme sur l'image ci-dessous l'adresse %I0.0. Même chose pour une sortie digitale, on pourra utiliser l'adresse %Q0.0.





Dans cette fiche, nous avons pu voir les variables automate et comment les adresser dans un programme.