



# R119 – Fonctions de base de la logique Séquentielle

Licence Pro Rob&IA

*Laurent ROY*



# Table des matières

- Partie I : Temporisations
- Partie II : Mémoires
- **Partie III : Compteurs**

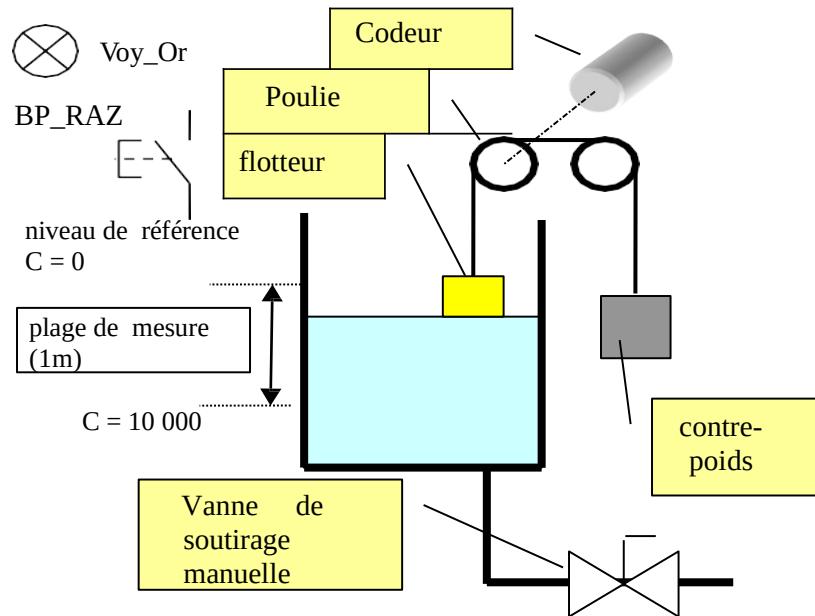
# III-1 Exemple introductif :

Mesure de niveau d'un réservoir : un flotteur équilibré par un contrepoids.

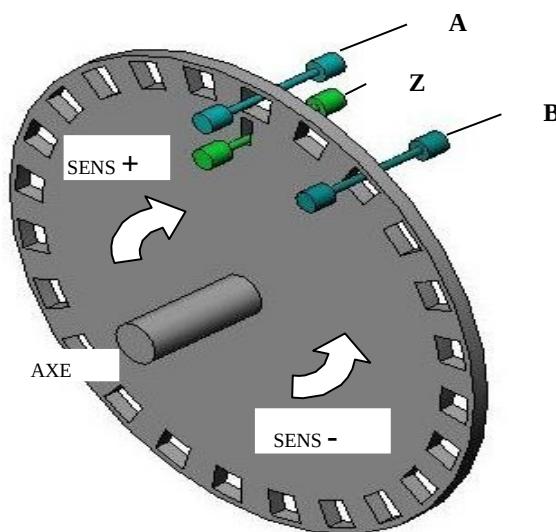
Un codeur incrémental **C** associé à la poulie permet de mesurer le niveau dans la cuve. Ce compteur compte de 0 à 10000 et est remis à "0" par un appui sur un bouton **BP\_RAZ**.

On suppose pour simplifier que la poulie effectue moins d'un tour sur la plage de mesure (1m).

Un voyant orange **Voy\_Or** est allumé quand le niveau de la cuve descend en dessous de 80 cm par rapport au niveau de référence.



## III-2 Codeur incrémental :



‘ n ’ est la **résolution** ou **nombre de points** : Pour un tour complet de l’axe du codeur le faisceau lumineux est interrompu ‘ n ’ fois et délivre ‘ n ’ signaux consécutifs.

2 diodes photosensibles décalées délivrant des signaux carrés **A** et **B**.

Signal **Z** : un ‘ Top zéro ’ par tour. Permet la réinitialisation à chaque tour.

- Pour connaître la valeur de la position angulaire de l’axe, il faut **compter les impulsions** en provenance du codeur.

## III-2 Codeur incrémental :

Les signaux en provenance du codeur ont les allures suivantes :

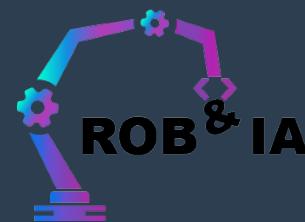


Les 2 signaux A et B sont décalés l'un par rapport à l'autre d'  $\frac{1}{4}$  de période.

→ permet de connaître le sens dans lequel tourne le codeur : Si quand A passe de "1" à "0" le signal B vaut "1", alors le sens est positif. Si B vaut "0" à cet instant, alors le sens de rotation est négatif. Il faut donc pouvoir détecter le front descendant de A pour connaître le sens de déplacement.

**NB :** Il existe un autre type de codeur, appelé "codeur absolu", qui sera vu en TD ultérieurement.

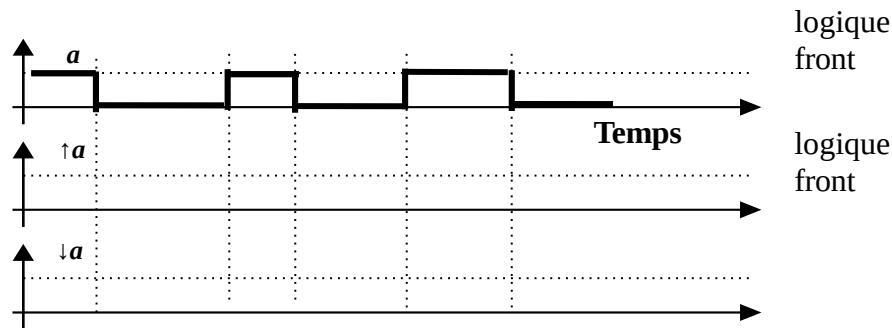
# III-3 FRONT MONTANT / FRONT DESCENDANT :



## 3.1 Définitions

A toute variable logique  $a$  peuvent être associées 2 variables logiques :

- l'une traduisant le passage de l'état "0" à l'état logique "1" : On l'appelle montant  $\uparrow a$
- l'une traduisant le passage de l'état "1" à l'état logique "0" : On l'appelle descendant  $\downarrow a$



Notation : en ladder :

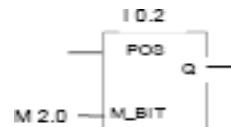
front montant :

front descendant :

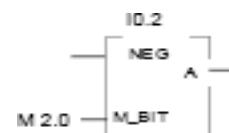
notation **LD**

en logigramme :

front montant :



front descendant :



notation **FBD**

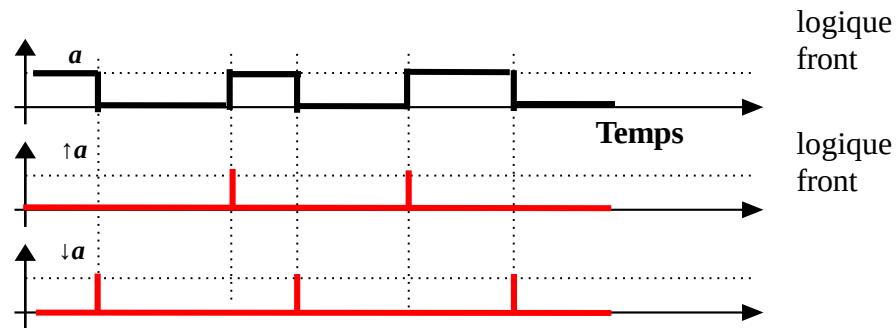
# III-3 FRONT MONTANT / FRONT DESCENDANT :



## 3.1 Définitions

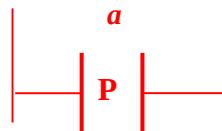
A toute variable logique  $a$  peuvent être associées 2 variables logiques :

- l'une traduisant le passage de l'état "0" à l'état logique "1" : On l'appelle montant  $\uparrow a$
- l'une traduisant le passage de l'état "1" à l'état logique "0" : On l'appelle descendant  $\downarrow a$

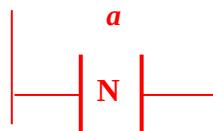


Notation : en ladder :

front montant :



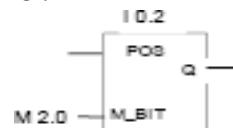
front descendant :



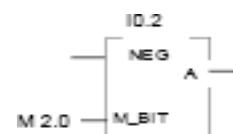
notation **LD**

en logigramme :

front montant :

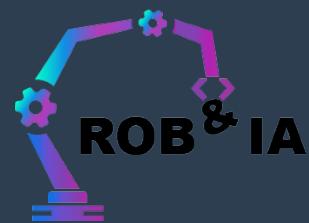


front descendant :



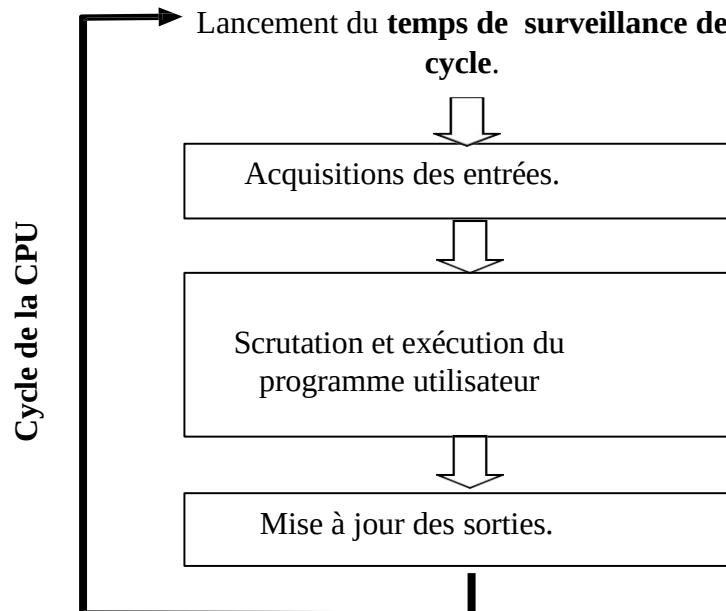
notation **FBD**

### III-3 FRONT MONTANT / FRONT DESCENDANT :



#### 3.2 Les fronts dans les programmes API

**Rappel :** L'exécution du programme de l'API se fait de manière cyclique quand l'automate est en mode "RUN".

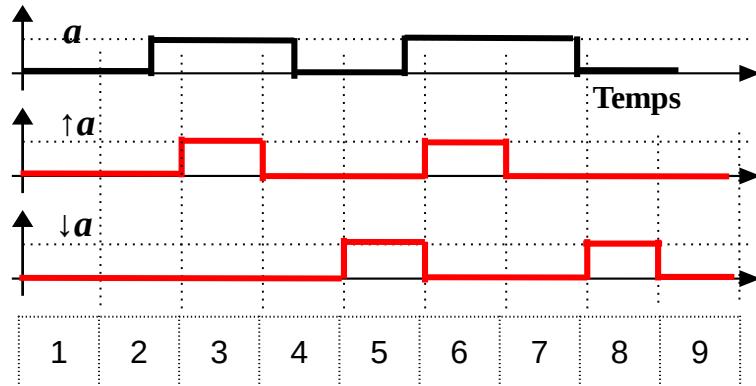
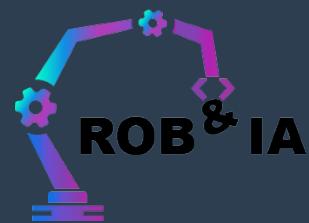


Ce cycle de CPU comprend 3 étapes :

- La CPU interroge l'état des signaux d'entrées.
- Elle exécute le programme utilisateur avec ses différentes opérations.
- Elle met à jour les sorties.

Le front montant théorique d'une variable est d'une durée nulle. Les fronts montant et descendant qui sont utilisés dans la programmation d'un automate restent à "1" pendant un temps de cycle de la CPU.

### III-3 FRONT MONTANT / FRONT DESCENDANT :



Cycles CPU Automate

Si un changement de "0" à "1" de la variable d'entrée  $a$  est détecté lors de la phase d'interrogation des entrées, alors la variable  $\uparrow a$  passe à "1". Au cycle suivant de CPU, la variable  $\uparrow a$  repasse à "0".

NB : Un cycle automate est de l'ordre de 10ms à 50ms

## III-4 Réalisation pratique de compteurs :



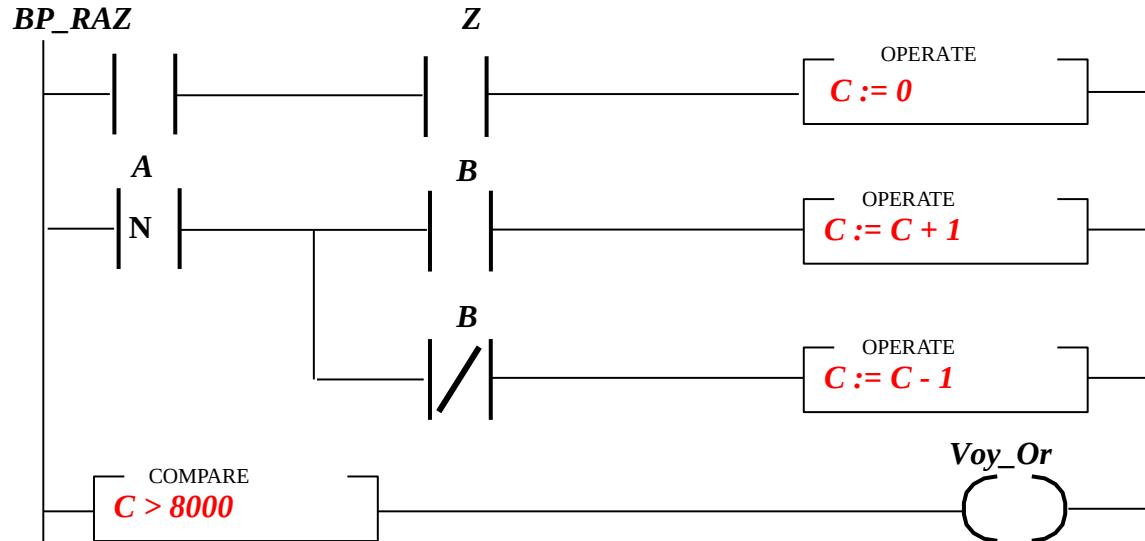
Quelle que soit la syntaxe adoptée, la programmation d'un compteur doit forcément comporter les 3 étapes suivantes :

- 1. Initialisation** du compteur
- 2. Incrémentation ou décrémentation sur front**
- 3. Test** de la valeur.

# III-4 Réalisation pratique de compteurs :

## 4.1 Compteur en LD

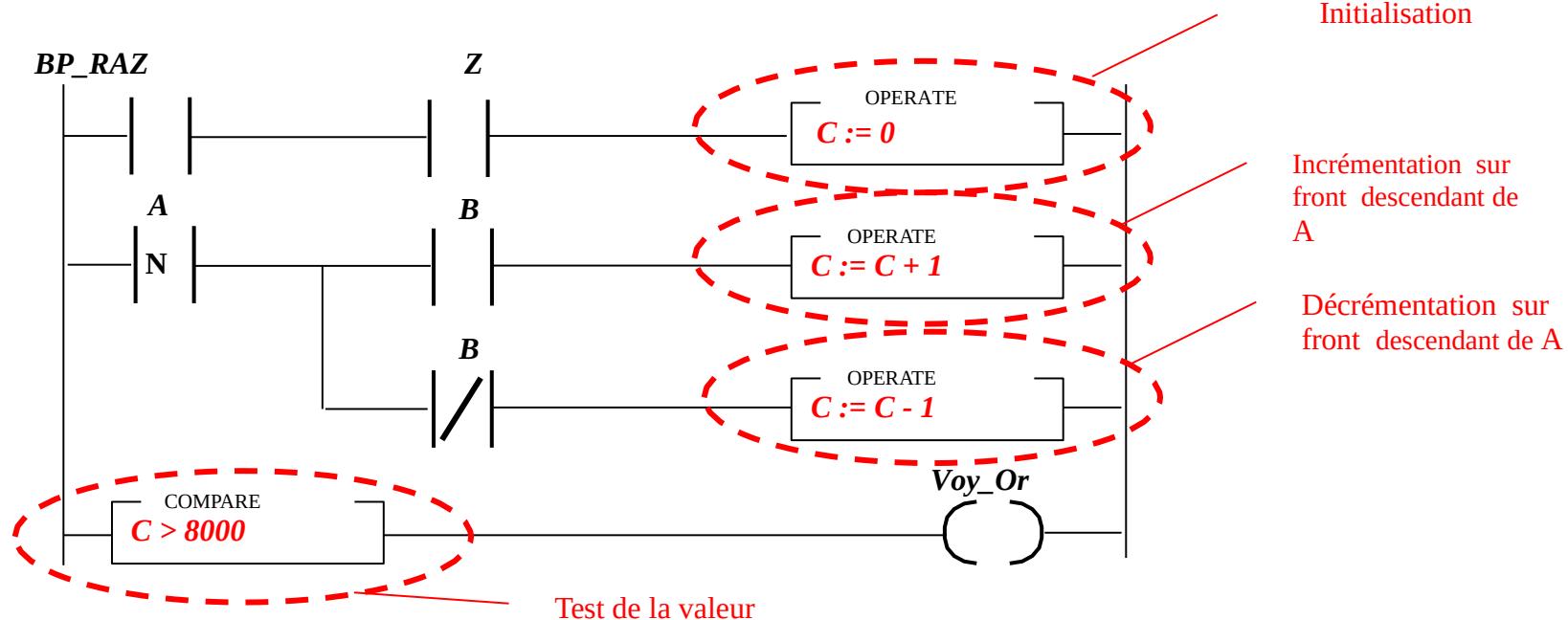
La programmation suivante en langage LD permet de répondre au cahier des charges défini au I/



# III-4 Réalisation pratique de compteurs :

## 4.1 Compteur en LD

La programmation suivante en langage LD permet de répondre au cahier des charges défini au I/

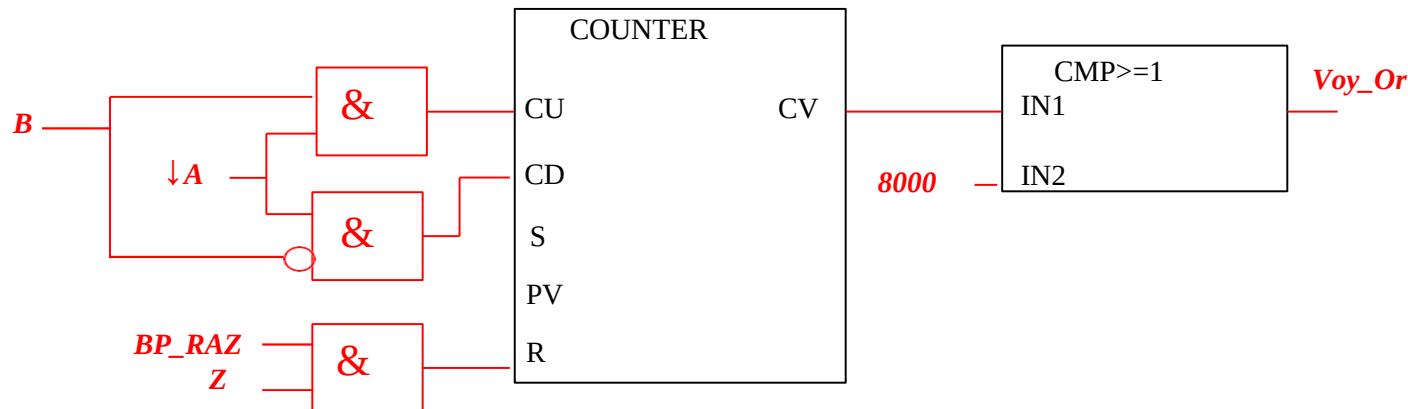


# III-4 Réalisation pratique de compteurs :



## 4.2 Compteur en FBD

La programmation suivante en langage FBD (Step7) permet de répondre au cahier des charges

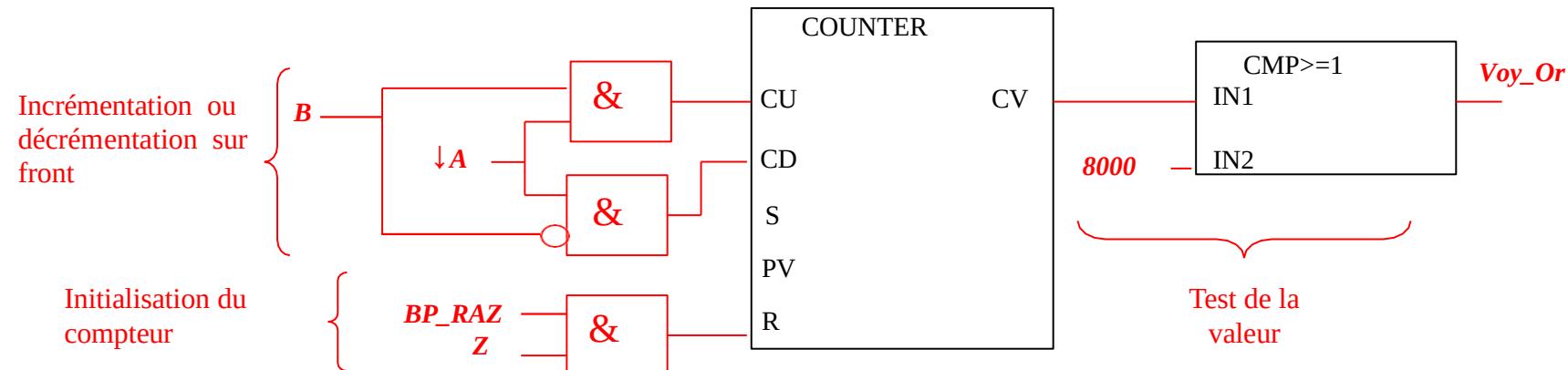


# III-4 Réalisation pratique de compteurs :



## 4.2 Compteur en FBD

La programmation suivante en langage FBD (Step7) permet de répondre au cahier des charges

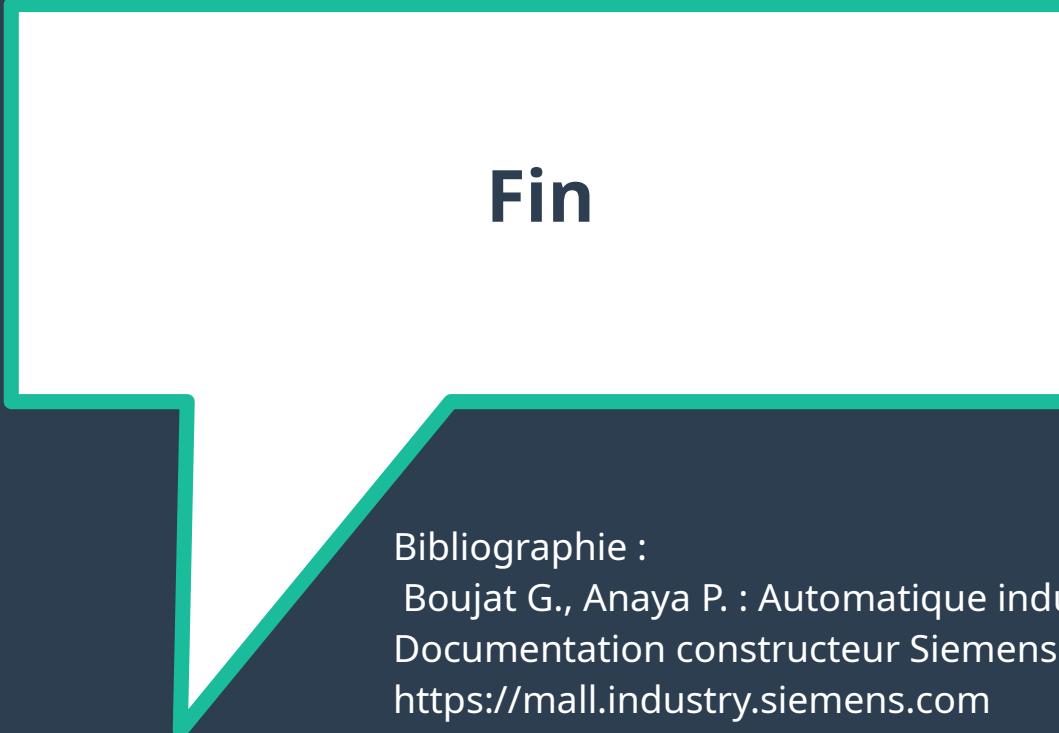


# III-4 Réalisation pratique de compteurs :



## 4.3 Modules de comptage

- Jusqu'à une fréquence de 500Hz, les signaux A, B et Z peuvent être raccordés directement sur les entrées TOR de l'API. Dans ce cas, **c'est le temps de cycle automatique qui limite cette fréquence de comptage.**
- Au-delà de cette fréquence, il faudra équiper l'automate **d'entrées de comptage rapide.**
- Certains automates sont équipés nativement d'entrées de comptage rapide (100kHz pour des entrées de comptage sur S71512c)



# Fin

Bibliographie :

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