

TP N°1
FONCTIONS LOGIQUES
DE BASE SUR TIA PORTAL V13

I. OBJECTIF

Réaliser les opérations logiques de base sur l'environnement SIEMENS TIA PORTAL V13.

II. MATERIEL

1 Micro-ordinateur avec conditions requises minimales :

- *Processeur => Intel® Celeron® Dual Core 2,2 GHz (Ivy/Sandy Bridge)*
- *RAM => 2 GB*
- *Espace disque disponible => 20 GB*
- *Systèmes d'exploitation => Windows 7 (64 bits)*
- *Résolution de l'écran => 1024 x 768*

III. MANIPULATIONS

1. Contact à fermeture --- | |--- :

1.1. Description

L'activation d'un contact à fermeture dépend de l'état logique de l'opérande correspondant. Quand l'opérande est à l'état logique "1", le contact à fermeture se ferme et l'état logique de l'entrée est transféré à la sortie. Quand l'opérande est à l'état logique "0", le contact à fermeture n'est pas activé et la sortie de l'instruction est mise à l'état logique "0".

Paramètre	Déclaration	Type de données	de Zone mémoire	Description
<opérande>	Input	BOOL	I, Q, M, D, L, T, C	Opérande dont l'état logique est interrogé.

- 1.1. **Travail demandé** : faites le programme Ladder de la figure 1 en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

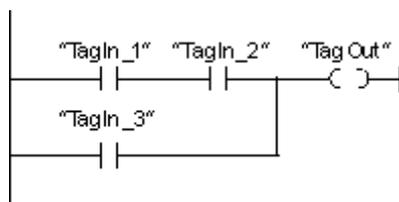


Figure 1- fonctionnement de l'instruction *Contact à fermeture*

2. Contact à ouverture ---| / |--- :

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
<opérande>	Input	BOOL	I, Q, M, D, L, T, C	Opérande dont l'état logique est interrogé.

2.1. Description

L'activation d'un contact à ouverture dépend de l'état logique de l'opérande correspondant. Quand l'opérande est à l'état logique "1", le contact à ouverture s'ouvre et la sortie de l'instruction est mise à l'état logique "0".

Quand l'opérande est à l'état logique "0", le contact à ouverture n'est pas activé et l'état logique de l'entrée est transféré à la sortie.

- 2.2. **Travail demandé** : faites le programme Ladder de la figure 2 en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

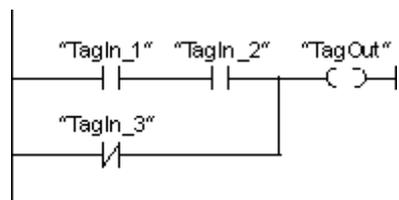


Figure 2- fonctionnement de l'instruction *Contact à ouverture*

3. Inverser RLO--|NOT|-- :

3.1. Description

L'instruction "Inverser RLO" permet d'inverser l'état logique du résultat logique (RLO). Lorsque l'état logique "1" se présente à l'entrée de l'instruction, sa sortie fournit l'état logique "0". Si l'état logique est "0" à l'entrée de l'instruction, sa sortie fournit l'état logique "1".

- 3.2. **Travail demandé** : faites le programme Ladder de la figure 3 en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

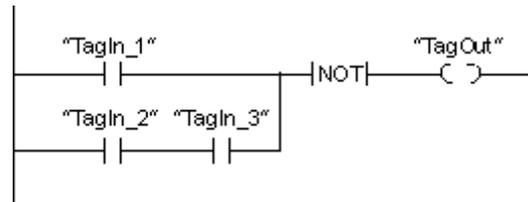


Figure 3- fonctionnement de l'instruction *NOT*

4. Affectation ---()--- :

4.1. Description

L'instruction "Affectation" permet de mettre à 1 le bit d'un opérande spécifié. Quand le résultat logique (RLO) à l'entrée de la bobine fournit l'état logique "1", l'opérande spécifié est mis à l'état logique "1". Lorsque l'état logique à l'entrée de la bobine est "0", le bit de l'opérande spécifié est remis à "0".

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
<opérande>	Output	BOOL	I, Q, M, D, L	Opérande auquel le RLO est affecté.

4.2. Travail demandé : faites le programme Ladder de la figure 4 en réalisant toutes les combinaisons possibles de TagIn1, TagIn2, TagIn3 et TagIn4.

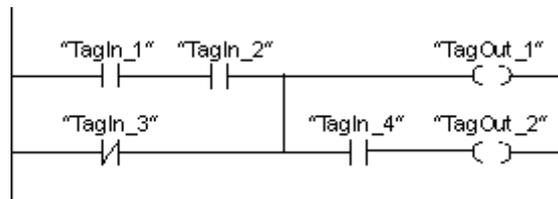


Figure 4- fonctionnement de l'instruction *Affectation*

5. Mise à 0 sortie ---(R)--- :

5.1. Description

L'instruction "Mise à 0 sortie" vous permet de remettre à "0" l'état logique d'un opérande spécifié.

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
<opérande>	Output	BOOL	I, Q, M, D, L, T, C	Opérande qui est remis à 0 lorsque le RLO = "1".

5.2. Travail demandé : faites le programme Ladder de la figure 5 en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

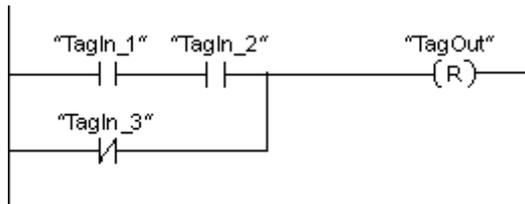


Figure 5- fonctionnement de l'instruction ---(R)---

6. Mise à 1 sortie ---(S)--- :

L'instruction "Mise à 1 sortie" vous permet de mettre à "1" l'état logique d'un opérande spécifié.

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
<opérande>	Output	BOOL	I, Q, M, D, L	Opérande mis à 1 quand RLO = "1".

- 6.1. **Travail demandé** : faites le programme Ladder de la figure 6 en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3 .

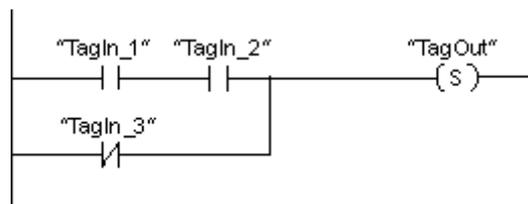


Figure 6- fonctionnement de l'instruction ---(S)---

7. Bascule SR : 'mise à 1/mise à 0'

L'instruction "Bascule 'mise à 1/mise à 0'" permet de mettre à 1 ou à 0 le bit d'un opérande spécifié quel que soit l'état logique aux entrées S et R1. Quand l'entrée S est à "1" et l'entrée R1 à "0", l'opérande spécifié est mis à "1". Quand l'entrée S est à "0" et l'entrée R1 à "1", l'opérande spécifié est remis à "0".

L'entrée R1 a la priorité sur l'entrée S. Quand les deux entrées S et R1 sont à "1", l'opérande spécifié est mis à "0".

Quand les deux entrées S et R1 sont à "0", l'instruction n'est pas exécutée. Dans ce cas, l'état logique de l'opérande ne change pas.

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
S	Input	BOOL	I, Q, M, D, L	Valider mise à 1
R1	Input	BOOL	I, Q, M, D, L, T, C	Valider mise à 0
<opérande>	InOut	BOOL	I, Q, M, D, L	Opérande à mettre à 1 ou à 0.
Q	Output	BOOL	I, Q, M, D, L	Etat logique de l'opérande

- 7.1. **Travail demandé** : faites le programme Ladder de la figure 7, en réalisant toutes les combinaisons possibles de TagIn1 et TagIn2.

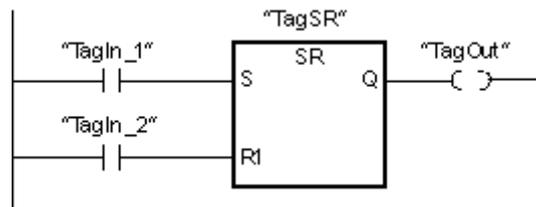


Figure 7- fonctionnement de l'instruction *Bascule SR*

8. RS : Bascule 'mise à 0/mise à 1'

8.1. Description

L'instruction "Bascule 'mise à 0/mise à 1'" vous permet de mettre à 0 ou à 1 le bit d'un opérande spécifié en fonction de l'état logique aux entrées R et S1. Quand l'entrée R est à "1" et l'entrée S1 à "0", l'opérande spécifié est remis à "0". Quand l'entrée R est à "0" et l'entrée S1 à "1", l'opérande spécifié est mis à "1".

L'entrée S1 a la priorité sur l'entrée R. Quand l'état logique est "1" aux deux entrées R et S1, l'état logique de l'opérande indiqué est mis à "1".

Quand les deux entrées R et S1 sont à "0", l'instruction n'est pas exécutée. Dans ce cas, l'état logique de l'opérande ne change pas.

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
R	Input	BOOL	I, Q, M, D, L	Valider mise à 0
S1	Input	BOOL	I, Q, M, D, L, T, C	Valider mise à 1
<opérande>	InOut	BOOL	I, Q, M, D, L	Opérande qui est mis à 0 ou à 1.
Q	Output	BOOL	I, Q, M, D, L	Etat logique de l'opérande

- 8.2. **Travail demandé** : faites le programme Ladder de la figure 8, en réalisant toutes les combinaisons possibles de TagIn1 et TagIn2.

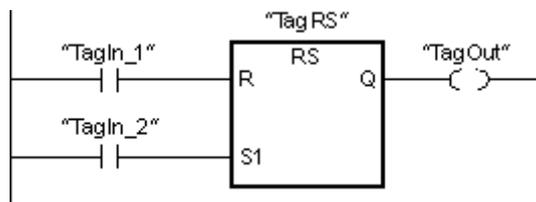


Figure 8- fonctionnement de l'instruction *Bascule RS*

9. Interroger front montant d'un opérande --|P|-- :

L'instruction "Interroger front montant d'un opérande" permet de détecter un changement de "0" à "1" dans l'état logique d'un opérande spécifié (<opérande1>). L'instruction compare l'état logique actuel de l'<opérande1> à celui de l'interrogation précédente, mémorisé dans un memento de front (<opérande2>). Quand l'instruction détecte un changement de "0" à "1" dans le (RLO), il s'agit d'un front montant.

Lorsqu'un front montant est détecté, la sortie de l'instruction fournit l'état logique "1". Dans tous les autres cas, l'état logique est "0" à la sortie de l'instruction.

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
<opérande1>	Input	BOOL	I, Q, M, D, L, T, C	Signal à interroger
<opérande2>	InOut	BOOL	I, Q, M, D, L	Mémento de front dans lequel est mémorisé l'état logique de l'interrogation précédente.

- 9.1. **Travail demandé** : faites le programme Ladder de la figure 9, en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

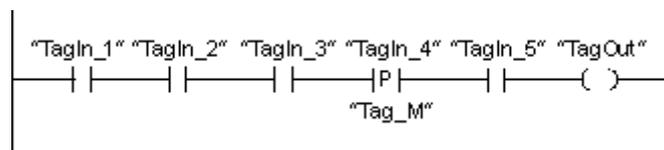


Figure 9- fonctionnement de l'instruction --|P|--

10. Mise à 1 de l'opérande si front montant du signal --(P)-- :

Lorsqu'un front montant est détecté, l'<opérande1> est mis à l'état logique "1" pendant un cycle du programme. Dans tous les autres cas, l'opérande fournit l'état logique "0".

Paramètres	Déclaration	Type de données	Zone de mémoire	Description
<opérande1>	Output	BOOL	I, Q, M, D, L	Opérande mis à 1 en cas de front positif.
<opérande2>	InOut	BOOL	I, Q, M, D, L	Mémento de front

- 10.1. **Travail demandé** : faites le programme Ladder de la figure 10, en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

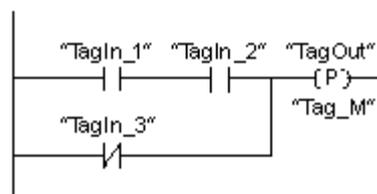


Figure 10- fonctionnement de l'instruction --(P)--

11. Mise à 1 de l'opérande si front descendant du signal--(N)-- :

L'instruction "Mise à 1 de l'opérande si front descendant du signal" permet de mettre un opérande spécifié à "1" (<opérande1>) quand le résultat logique (RLO) passe de "1" à "0". Cette instruction compare le RLO actuel au RLO de l'interrogation précédente qui est

mémorisé dans un memento de front (<opérande2>). Quand l'instruction détecte un changement de "1" à "0" dans le (RLO), il s'agit d'un front descendant.

Lorsqu'un front descendant est détecté, l'<opérande1> est mis à l'état logique "1" pendant un cycle du programme. Dans tous les autres cas, l'opérande fournit l'état logique "0".

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
<opérande1>	Output	BOOL	I, Q, M, D, L	Opérande mis à 1 en cas de front descendant.
<opérande2>	InOut	BOOL	I, Q, M, D, L	Memento de front

11.1. Travail demandé : faites le programme Ladder de la figure 11, en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

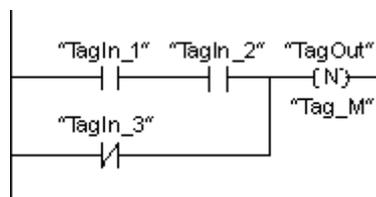


Figure 11- fonctionnement de l'instruction --(N)--

12. Interroger front montant du RLO P_TRIG :

L'instruction "Interroger front montant du RLO" permet d'interroger le passage de "0" à "1" du résultat logique (RLO). Cette instruction compare l'état logique actuel du RLO à celui de l'interrogation précédente, mémorisé dans un memento de front (<opérande>). Quand l'instruction détecte un changement de "0" à "1" dans le RLO, il s'agit d'un front montant.

Lorsqu'un front montant est détecté, la sortie de l'instruction fournit l'état logique "1". Dans tous les autres cas, l'état logique est "0" à la sortie de l'instruction.

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
CLK	Input	BOOL	I, Q, M, D, L	RLO actuel
<opérande>	InOut	BOOL	M, D	Memento de front dans lequel est mémorisé le RLO de l'interrogation précédente.
Q	Output	BOOL	I, Q, M, D, L	Résultat de l'évaluation du front

12.1. Travail demandé : faites le programme Ladder de la figure 12, en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

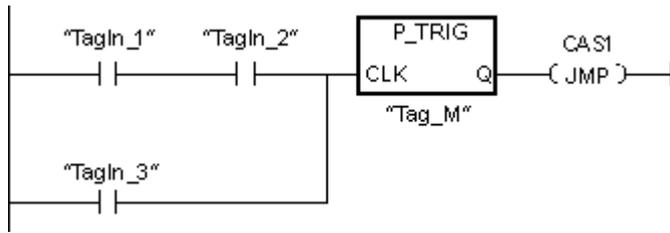


Figure 12- fonctionnement de l'instruction **P_TRIG**

13. Mise à 1 de la variable si front montant du signal R_TRIG (s7-1200/s7-1500)

Avec l'instruction "Mise à 1 de la variable si front montant du signal", vous pouvez mettre une variable spécifiée à "1" dans le DB d'instance quand le résultat logique (RLO) passe de "0" à "1". Cette instruction compare le RLO actuel à l'entrée CLK au RLO de l'interrogation précédente qui est mémorisé dans le DB d'instance spécifié. Quand l'instruction détecte un changement de "0" à "1" dans le RLO, il s'agit d'un front montant.

Lorsqu'un front montant est détecté, la variable dans le DB d'instance est mise à l'état logique "1" et la sortie Q fournit l'état logique "1". Dans tous les autres cas, l'état logique est "0" à la sortie de l'instruction.

Paramètres	Déclaration	Type de données	Zone de mémoire	Description
EN	Input	BOOL	I, Q, M, D, L	Entrée de validation
ENO	Output	BOOL	I, Q, M, D, L	Sortie de validation
CLK	Input	BOOL	I, Q, M, D, L ou constante	Signal entrant dont le front est interrogé.
Q	Output	BOOL	I, Q, M, D, L	Résultat de l'évaluation du front

13.1. Travail demandé : faites le programme Ladder de la figure 13, en réalisant toutes les combinaisons possibles de TagIn1, TagIn2 et TagIn3.

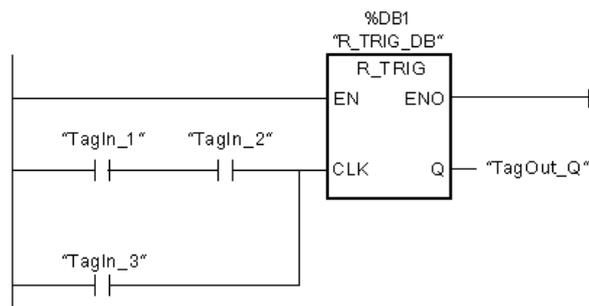


Figure 13- fonctionnement de l'instruction **R_TRIG**