



1. Les fonctions logiques

| CONT | LOG | SCL | Commentaires |
|------|-----|---------------------|---|
| | | out := in1 AND in2; | Toutes les entrées d'une boîte ET doivent être vraies pour que la sortie soit vraie. |
| | | out := in1 OR in2; | Il suffit qu'une entrée d'une boîte OU soit vraie pour que la sortie soit vraie. |
| | | out := in1 XOR in2; | Un nombre impair d'entrées d'une boîte OU EXCLUSIF doivent être vraies pour que la sortie soit vraie. |
| | | NOT | <p>En LOG, vous pouvez faire glisser l'outil "Inverser l'entrée binaire" de la barre d'outils "Favoris" ou de l'arborescence d'instructions vers une entrée ou une sortie afin de créer un inverseur logique sur ce connecteur de boîte</p> <p>En CONT « NOT » inverse l'état logique</p> |

2. L'affectation

| CONT | LOG | SCL | Commentaires |
|------|-----|---|---|
| | | out := <expression booléenne>; | En LOG , les bobines CONT sont transformées en boîte d'affectation (= et /=) dans lesquelles vous indiquez une adresse de bit pour la sortie de la boîte. |
| | | out := NOT <expression booléenne>; | Les entrées et sorties de la boîte peuvent être connectées à une autre boîte logique ou vous pouvez entrer une adresse de bit. |
| | 0 | L'instruction "Inverser RLO" permet d'inverser l'état logique du résultat logique (RLO). | |
| | /= | Négation de l'affectation L'instruction "Négation de l'affectation" inverse le résultat logique (RLO) et l'affecte à l'opérande se trouvant au-dessus de la boîte. | |

3. Mise à « 0 » ou à « 1 »

| CONT | LOG | SCL | Commentaires |
|------|-----|----------------|--|
| | | Pas disponible | <p>L'instruction "Mise à 1 sortie" vous permet de mettre à "1" l'état logique d'un opérande spécifié.</p> <p>L'instruction n'est exécutée que si le résultat logique (RLO) à l'entrée de la boîte est égal à "1". Pour un RLO égal à "0" à l'entrée de la boîte, l'état logique de l'opérande spécifié reste inchangé.</p> |



| | | | |
|--|--|----------------|--|
| | | Pas disponible | <p>L'instruction "Mise à 0 sortie" vous permet de remettre à "0" l'état logique d'un opérande spécifié.</p> <p>L'instruction n'est exécutée que si le résultat logique (RLO) à l'entrée de la boîte est égal à "1". Pour un RLO égal à "0" à l'entrée de la boîte, l'état logique de l'opérande spécifié reste inchangé.</p> |
|--|--|----------------|--|

4. Détection de transition front montant ou descendant d'un opérande

Détecter une transition (un front) permet de prendre en compte une information sur un seul cycle.

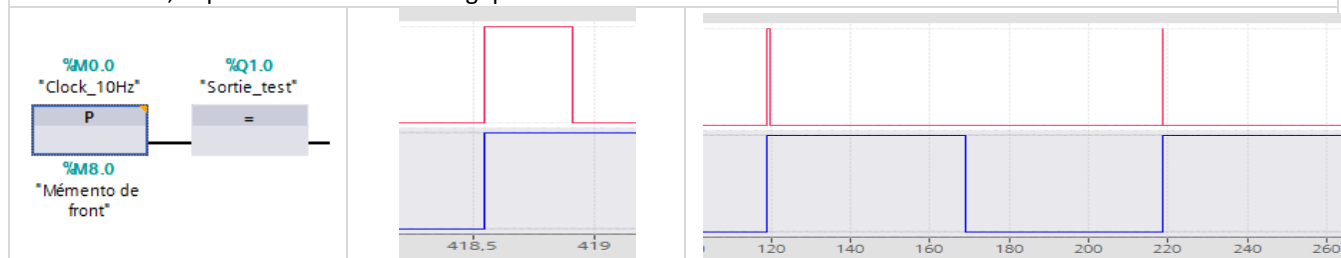
Cela s'avère très utile pour des opérations de comptage et de détection de passage.

| CONT | LOG | SCL | Commentaires |
|------|-----|----------------|---|
| | | Pas disponible | <p>CONT : L'état de ce contact est VRAI lorsqu'un front montant (0 à 1) est détecté sur l'opérande affecté. Le contact P peut être situé n'importe où dans le réseau, excepté à la fin d'une branche.</p> <p>LOG : L'état logique de la sortie est VRAI lorsqu'un front montant (0 à 1) est détecté sur l'opérande affecté.</p> <p> La boîte P ne peut être située qu'au début d'une branche.</p> |

Le memento sera déclaré dans une DB d'instance (en zone statique) ou dans la zone des mementos des variables API (mémoire M). On veillera à ce que l'état de ce memento ne soit pas écrasé en un autre point du programme en ayant soin d'utiliser un memento de mémoire séparé pour chaque évaluation de front.

L'instruction d'évaluation compare l'état courant de l'opérande « opérande » à l'état qu'il avait lors du cycle de traitement précédent. Cet état est mémorisé à cet effet dans un memento auxiliaire « memento ».

En cas de changement d'état, l'instruction fournit de résultat logique « 1 » pendant un cycle du programme. Dans tous les autres cas, l'opérande fournit l'état logique « 0 ».

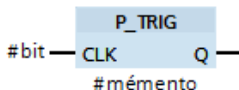


Le diagramme temporel illustre un changement d'état sur flanc montant(en rouge) et l'agrandissement de ce changement nous donne un temps de cycle de 400us.

| | | | |
|--|--|----------------|---|
| | | Pas disponible | <p>CONT : L'état de ce contact est VRAI lorsqu'un front descendant (1 à 0) est détecté sur l'opérande affecté. Le contact N peut être situé n'importe où dans le réseau, excepté à la fin d'une branche.</p> <p>LOG : L'état logique de la sortie est VRAI lorsqu'un front descendant (1 à 0) est détecté sur l'opérande affecté.</p> <p> La boîte N ne peut être située qu'au début d'une branche.</p> |
|--|--|----------------|---|



5. Détection de transition front montant ou descendant d'un bit

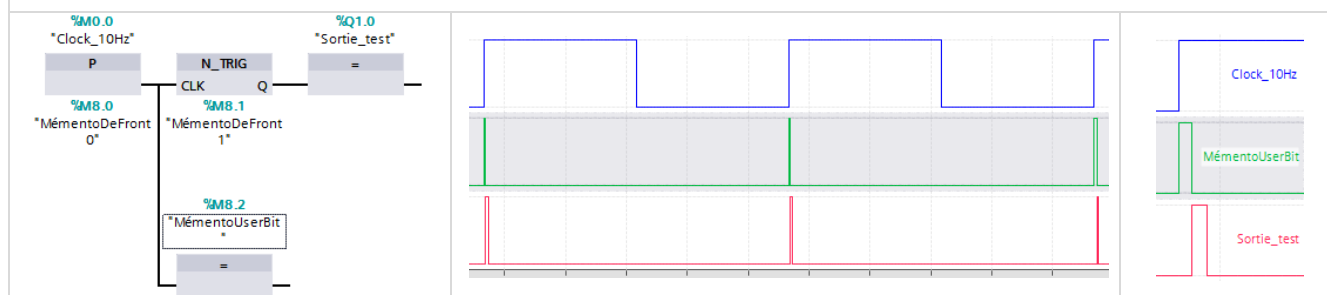
| CONT | LOG | SCL | Commentaires |
|---|-----|----------------|---|
|  | | Pas disponible | L'interrogation du front montant intervient à chaque exécution de l'instruction. La sortie Q de l'instruction fournit l'état logique "1" pendant un cycle du programme dès qu'un front montant est détecté. Dans tous les autres cas, la sortie fournit l'état logique "0". |

L'instruction d'évaluation compare l'état courant du bit d'entrée à l'état qu'il avait lors du cycle de traitement précédent. Cet état est mémorisé à cet effet dans un memento auxiliaire « memento».

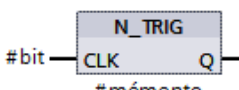
En cas de changement d'état, l'instruction fournit de résultat logique « 1 » pendant un cycle du programme. Dans tous les autres cas, l'opérande fournit l'état logique « 0 ».



Le memento sera déclaré dans une DB d'instance (en zone statique) ou dans la zone des mementos des variables API (mémoire M). On veillera à ce que l'état de ce memento ne soit pas écrasé en un autre point du programme en ayant soin d'utiliser un memento de mémoire séparé pour chaque évaluation de front.



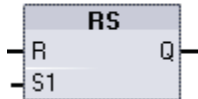

Le diagramme temporel illustre un changement d'état sur flanc montant(en vert) de l'opérande %M0.0 suivi par une détection du flanc descendant du bit présent à l'entrée du N_TRIG.

| | | |
|---|----------------|---|
|  | Pas disponible | L'interrogation du front descendant intervient à chaque exécution de l'instruction. La sortie Q de l'instruction fournit l'état logique "1" pendant un cycle du programme dès qu'un front descendant est détecté. Dans tous les autres cas, l'état logique est "0" à la sortie de l'instruction |
|---|----------------|---|

| | | | | | | | | | |
|-------------------------------|----|------------------|-------------------------|------|-------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Variables API | 13 | Clock_10Hz | Table de variables s... | Bool | %M0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Afficher toutes les vari... | 14 | Clock_5Hz | Table de variables s... | Bool | %M0.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Insérer une nouvelle ta... | 15 | Clock_2.5Hz | Table de variables s... | Bool | %M0.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Table de variables stan... | 16 | Clock_2Hz | Table de variables s... | Bool | %M0.3 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Types de données API | 17 | Clock_1.25Hz | Table de variables s... | Bool | %M0.4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Tables de visualisation et d. | 18 | Clock_1Hz | Table de variables s... | Bool | %M0.5 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Sauvegardes en ligne | 19 | Clock_0.625Hz | Table de variables s... | Bool | %M0.6 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Traces | 20 | Clock_0.5Hz | Table de variables s... | Bool | %M0.7 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Données d'appareil prox... | 21 | Entrée_3 | Table de variables s... | Bool | %I0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | 22 | mémoire SR1 | Table de variables s... | Bool | %M1.7 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Vue détaillée | 23 | Memento de front | Table de variables s... | Bool | %M8.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |



6. Bascule RS et SR

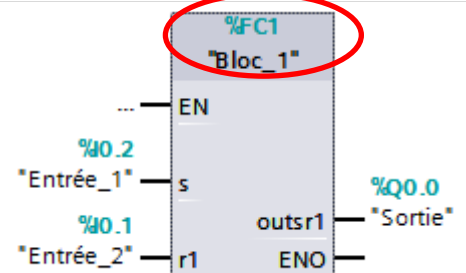
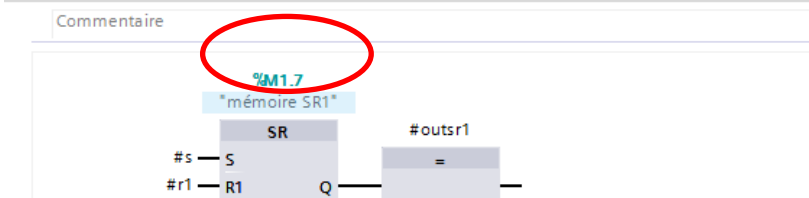
| CONT | LOG | SCL | Commentaires | | | | | | | | | | | | | | | |
|---|-----|----------------|---|-------|----|---|---|---|-------|---|---|---|---|---|---|---|---|---|
| <div>"OUT"</div> <div></div> | | Pas disponible | L'opération RS est une bascule où la mise à 1 domine . Si les signaux de mise à 1 (S1) et de mise à 0 (R) sont tous deux vrais, l'adresse de sortie OUT sera à 1. | | | | | | | | | | | | | | | |
| | | | <table><tr><th>S1</th><th>R</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>Q t-1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> | S1 | R | Q | 0 | 0 | Q t-1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| | | S1 | R | Q | | | | | | | | | | | | | | |
| | | 0 | 0 | Q t-1 | | | | | | | | | | | | | | |
| | | 0 | 1 | 0 | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| <div>"OUT"</div> <div></div> | | Pas disponible | L'opération SR est une bascule où la mise à 0 domine . Si les signaux de mise à 1 (S) et de mise à 0 (R1) sont tous deux vrais, l'adresse de sortie OUT sera à 0. | | | | | | | | | | | | | | | |
| | | | <table><tr><th>S</th><th>R1</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>Q t-1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table> | S | R1 | Q | 0 | 0 | Q t-1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| | | S | R1 | Q | | | | | | | | | | | | | | |
| | | 0 | 0 | Q t-1 | | | | | | | | | | | | | | |
| | | 0 | 1 | 0 | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |

*

je découvre > PLC_1 [CPU 1214C DC/DC/DC] > Blocs de programme > Bloc_1 [FC1]



| | Nom | Type de données | Valeur par déf. | Commentaire |
|---|----------|-----------------|-----------------|-------------|
| 1 | Input | | | |
| 2 | s | Bool | | |
| 3 | r1 | Bool | | |
| 4 | Output | | | |
| 5 | outsr1 | Bool | | |
| 6 | InOut | | | |
| 7 | Temp | | | |
| 8 | Constant | | | |
| 9 | Return | | | |

Titre du bloc : ...
CommentaireRéseau 1 : ...
Commentaire

Première solution :

Construire une FC

L'état de la bascule est stocké dans une mémoire globale.

Dans l'exemple ci-contre : » %M1.7 »



Deuxième solution :

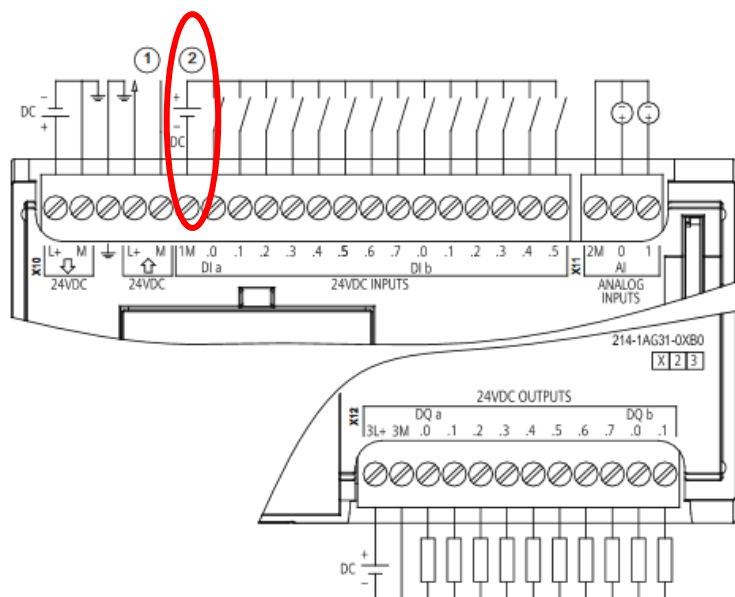
Construire une FB avec une DB d'instance pour y stocker l'état de la bascule en variable statique.

7. Etats logiques des entrées

| | |
|--|---|
| | <p>Contact normalement ouvert NO sera considéré comme un état logique « 0 » au repos</p> <p>Contact normalement fermé NF sera considéré comme un état logique « 1 » au repos</p> |
| | <p>Les contacts de fin de course peuvent être NO ou NF</p> <p>L'arrêt d'urgence est toujours un NF</p> |
| | <p>Sortie du capteur NPN</p> <p>La charge (bobine de relais, entrée de contacteur auxiliaire, entrée automate) est à installer entre le + et le point milieu.</p> <p>L'entrée automate correspondante sera Source</p> <p>Le PLC détectera V- (souvent 0V) lors de l'activation du capteur.</p> |
| | <p>Sortie du capteur PNP</p> <p>La charge (bobine de relais, entrée de contacteur auxiliaire, entrée automate) est à installer entre le - et le point milieu.</p> <p>L'entrée automate correspondante sera Sink</p> <p>Le PLC détectera V+ (souvent 24V) lors de l'activation du capteur.</p> |



Ci-dessous, la documentation du S7 impose le raccordement de 1M à M (0V) pour faire fonctionner les entrées en mode Sink (PNP). A noter que toutes les entrées seront alors du type PNP.



- ① 24 VDC Sensor Power Out
For additional noise immunity, connect "M" to chassis ground even if not using sensor supply.
- ② For sinking inputs, connect "-" to "M" (shown).
For sourcing inputs, connect "+" to "M".
- Note: X11 connectors must be gold. See Appendix C, Spare Parts for order number.

8. Exercices

Concevez les schémas en CONT et en LOG des équations suivantes :

| | CONT | LOG |
|-----------------------|------|-----|
| $S = A/B + /C.D$ | | |
| $S = (A+/B+C) . /D+E$ | | |
| $S = /(A.B) + /(C+D)$ | | |
| $S = (A + (/B.C))$ | | |



Interprétation de logigramme

Etablir l'équation logique

Commenter cette fonction

Quel pourrait être le rôle de #in3 ?

Etablir la table vérité

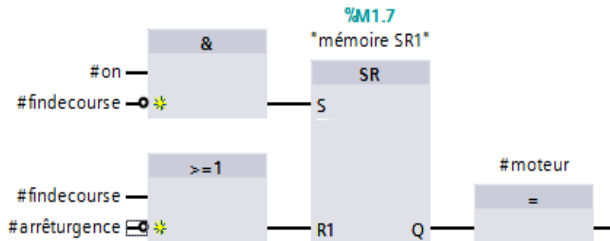
| | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Etablir la table vérité

| | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |



Commentez le logigramme de pilotage d'un moteur

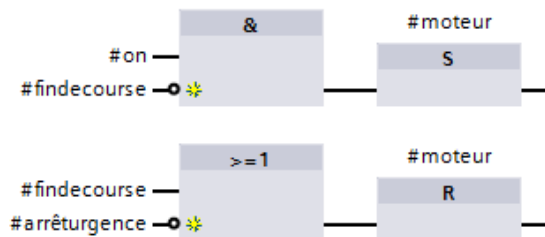


#moteur est à « 1 » dans quelles conditions ?

#moteur est à « 0 » dans quelles conditions ?

Quelles sont les signaux prioritaires ?

Identifiez les variables locales et globales.



Comparez les deux logigrammes.

Deux contacteurs « Start », « Stop » et un capteur de présence.

Niveau 1

Si on enfonce « Start » en présence d'une pièce sur un convoyeur, celui-ci démarre.

Si on enfonce « Stop », le tapis s'arrête

Niveau 2

Le même exercice en remplaçant les contacteurs par des boutons poussoirs.



| Deux boutons poussoirs « Start », « Stop » et deux capteurs de présence. | |
|--|---|
| | <p>Niveau 1</p> <p>Le convoyeur démarre lorsqu'une pièce est présente à l'entrée et que « Start » est enfoncé.</p> <p>Le convoyeur s'arrête si « Stop » est enfoncé ou présence d'une pièce devant le deuxième capteur.</p> |
| | <p>Niveau 2</p> <p>Le même exercice en ajoutant de la signalisation à 1 Hz et un arrêt d'urgence</p> |

| Deux poussoirs « UP », « Down » actionne un vérin double effet. | |
|---|--|
| | <p>Niveau 1 :</p> <p>Une action sur un poussoir actionne le vérin dans un sens opposé au sens précédent.</p> |
| | <p>Niveau 2 :</p> <p>Le vérin est équipé de deux contacteurs de fin de course. L'action sur le poussoir ne peut être prise en compte que si le vérin est en butée.</p> |



Une pompe est commandée grâce à deux poussoirs Marche, Arrêt.

Le système automatisé comporte les éléments suivants :

| | |
|------------------------|--------|
| 1 moteur de pompe | P |
| Un poussoir « Marche » | Marche |
| Un poussoir « Arrêt » | Arrêt |
| Un Arrêt d'urgence | AU |

Niveau 1 :

Si le poussoir « Marche » est enfoncé et que la pompe est arrêtée, la pompe démarre. Elle continue de tourner si le poussoir est relâché.

Si le poussoir « Arrêt » est enfoncé et que la pompe tourne, la pompe s'arrête. Elle reste arrêtée si le poussoir est relâché.

Niveau 2 :

Un arrêt d'urgence doit être ajouté.

Niveau 3 :

Prévoir le cas suivant :

Les deux poussoirs sont enfoncés.

L'opérateur relâche le poussoir d'arrêt.

Imaginez deux solutions différentes pour que la pompe reste à l'arrêt



Deux modes de fonctionnement pour un convoyeur. Le mode « Automatique » est prioritaire.

Le système automatisé comporte les éléments suivants :

| | |
|--------------------------------|----------|
| 1 moteur de convoyeur | M |
| Un contacteur « Automatique ». | Auto |
| Un contacteur « Manuel ». | Manu |
| 2 fins de course | FCG, FCD |

Niveau 1 :

Le mode « Automatique »

Dès que la présence d'une pièce est détectée à l'entrée du convoyeur, celui-ci démarre pour s'arrêter à l'arrivée de la pièce à l'autre extrémité.

Niveau 2 :

Ajouter le mode « Manuel »

« Start » démarre le convoyeur. Le convoyeur s'arrête par action sur « Stop » ou par le capteur de fin de convoyeur.

Niveau 3 :

Ajouter une signalisation lente à l'arrêt et rapide en fonctionnement du convoyeur.



Gestion d'une porte de garage.

Le système automatisé comporte les éléments suivants :

| | |
|--|----------|
| 1 moteur d'ouverture /fermeture | M |
| 1 vérin de verrouillage ouvert/fermé | V |
| 2 boutons poussoirs ouverture/fermeture | SO, SF |
| 2 fins de course porte ouverte, porte fermée | FCF, FCO |

Niveau 1 :

Une action SO démarre le processus d'ouverture
Le vérin déverrouille la porte.
Le déverrouillage accompli, la porte s'ouvre.
FCO signale la fin de l'ouverture.
Le vérin verrouille la porte.

Une action SF démarre le processus de fermeture
Le vérin déverrouille la porte.
Le déverrouillage accompli, la porte se ferme.
FCF signale la fin de la fermeture.
Le vérin verrouille la porte.

Niveau 2 :

Ajouter une signalisation lumineuse.

Vert : porte au repos
Rouge : porte en action