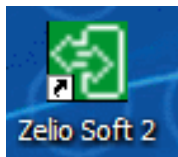




Programmer un automate

Programmation ZELIO



jeudi 27 juillet 2017

Connaissances	N°	Capacités à acquérir	NT *
Traitement du signal : algorithme, organigramme, programme.	5.5	Identifier les étapes d'un programme de commande représenté sous forme graphique.	1
Traitement du signal : algorithme, organigramme, programme.	5.6	Modifier la représentation du programme de commande d'un système pour répondre à un besoin particulier et valider le résultat obtenu.	2
Commande d'un objet technique et logique combinatoire de base : ET, OU, NON.	5.7	Identifier une condition logique de commande.	2

BO ou Référentiel : BO spécial n°6 du 28 Août 2008

* NT : Niveau Taxonomique (1 : Information / 2 : Expression / 3 : Maîtrise d'outils)

SITUATION DANS L'ANNEE :	• /
PREREQUIS :	• Les fonctions logiques, Barrière automatique
DUREE :	• 1 séance de 1 heure 30
SUPPORTS :	
DOCUMENTS :	• Document réponse élève • TPWorks
AUDIO-VISUELS :	• /
AUTRES :	• Logiciel ZELIO • Matériel - Voir page 2/2
BIBLIOGRAPHIE :	• /
LIENS :	• /

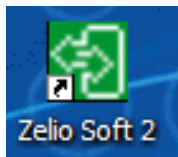
	N°	Type	Intitulé	Support	Conn.	Durée
Activités Séance	1	Expérimentation	Travail sur poste informatique Découvrir l'automate et configurer le logiciel	TPWorks	5.5 5.6	10 mn
	2	Expérimentation	Travail sur poste informatique Programmer un exemple simple en langage LADDER	TPWorks	5.5 5.6 5.7	30 mn
	3	Expérimentation	Travail sur poste informatique Programmer un exemple simple en langage FBD	TPWorks	5.5 5.6 5.7	20 mn
	4	Expérimentation	Travail sur poste informatique Programmer en langage FBD la levée de la barrière et tester sur maquette réelle le programme réalisé.	TPWorks Platine d'essai et matériel de câblage	5.5 5.6 5.7	30 mn
	5	Synthèse	Travail en classe entière Synthèse	Vidéoprojecteur		15 mn

Comment un portail, une barrière ou une porte de garage peuvent-elles s'ouvrir sans intervention de l'homme ?

4

Programmer un automate

Programmation ZELIO



jeudi 27 juillet 2017

Préparation Matériel / Ilot

		
Maquette de Barrière	Alimentation - 2 Supports de piles LR6 - 1,5V	Boutons poussoirs et boutons poussoirs à accrochage
		
Bornier double	Fil souple	Capteurs fin de course à galet
		
Platine Automate	Câble USB	Logiciel Zélio Soft installé

Comment un portail, une barrière ou une porte de garage peuvent-elles s'ouvrir sans intervention de l'homme ?

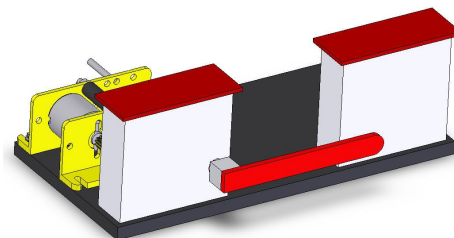
Programmation ZELIO

Page 1/6

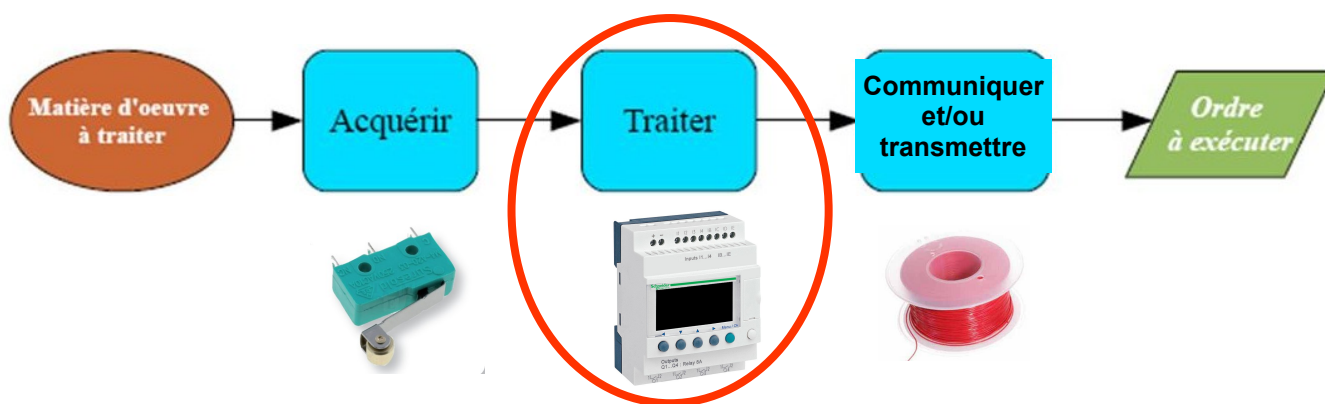
Présentation de l'activité

L'ouverture ou la fermeture de la barrière a pu être obtenu grâce à la réalisation d'un circuit électrique câblé.

Cependant, si on souhaite multiplier les options (Installer un gyrophare, un détecteur de présence, un digicode ...), le système devient très complexe et nécessite un « ordinateur » spécifique appelé **automate programmable**.

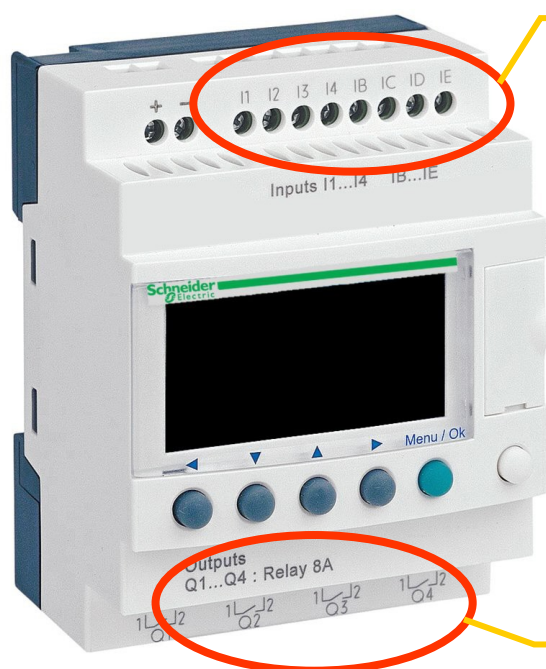


L'automate **ZELIO** en notre possession, cerveau de la chaîne d'information, **traite les informations acquises par les détecteurs et capteurs et transmet les ordres à exécuter** via les câbles électriques.



Mais pour pouvoir fonctionner, **il faut programmer l'automate** ...

Pour réaliser un programme, on utilise le logiciel **ZELIOSOFT** installé sur votre poste informatique. Une fois ce programme réalisé, on le télécharge dans la mémoire de l'automate et on l'exécute.



Entrées

Notre automate comporte **8 entrées** (appelées I1, I2, I3, I4, IB, IC, ID, IE)

On peut recevoir des informations provenant de **8 capteurs ou détecteurs**.

Il comporte **4 sorties** (appelées Q1, Q2, Q3, Q4)

On peut transmettre **4 ordres**.

Sorties

Première partie : Mettre en œuvre le logiciel ZELIOSOFT - Langage LADDER

Le logiciel **ZELIOSOFT** peut permettre la programmation de différents types d'automate de la gamme **ZELIO**. Il faut donc au préalable **le configurer** en fonction du modèle disponible dans le laboratoire de technologie mais aussi du mode de programmation souhaité.

1.1 Pour démarrer le logiciel, double cliquer sur l'icône du bureau



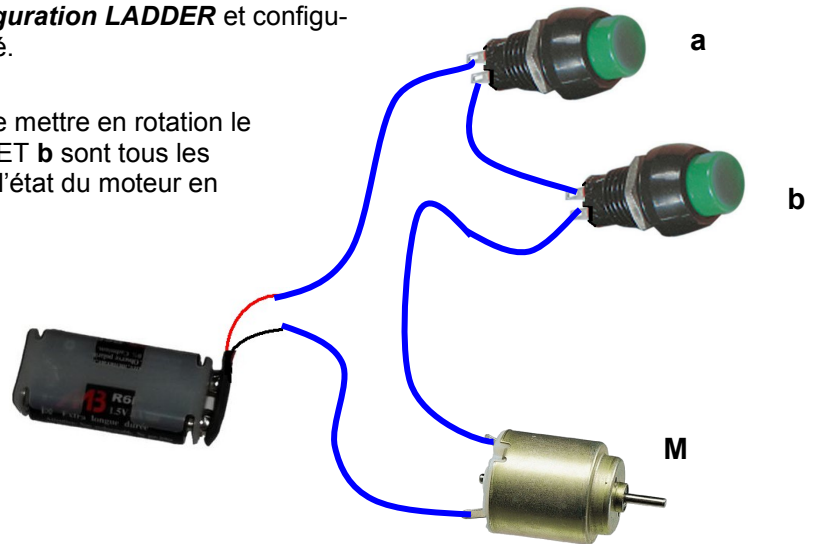
Il existe deux types de programmation :

- *Le langage **LADDER** s'apparente au schéma électrique.*
- *Le langage **FBD** permet entre autres une représentation en schéma logique (Fonctions Et, OU ...)*

1.2 Regarder la **Vidéo 01 - ZELIO - Configuration LADDER** et configurer le logiciel ZELIOSOFT comme présenté.

1.3 Le circuit électrique ci-contre permet de mettre en rotation le moteur **M** si les deux boutons poussoirs **a** ET **b** sont tous les deux appuyés. La table de vérité confirme l'état du moteur en fonction de l'état des boutons.

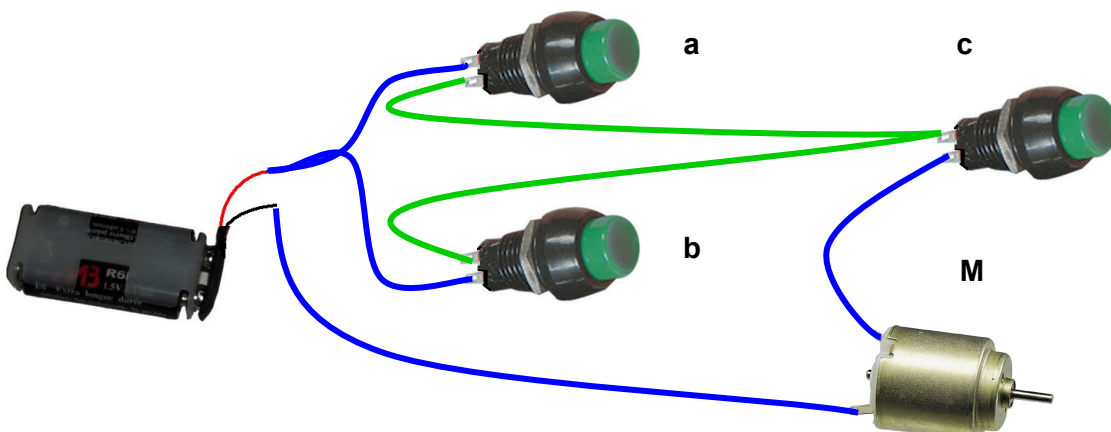
a	b	M
0	0	0
0	1	0
1	1	1
1	0	0



- Regarder la **Vidéo 02 - ZELIO - Programmation LADDER**
- Réaliser le programme correspondant au circuit électrique présenté dans la vidéo
- Vérifier les résultats de la table de vérité
- Faire vérifier votre travail par le professeur.

1.4 Soit le circuit électrique ci-dessous.

- Réaliser le programme en langage LADDER.
- Compléter la table de vérité.
- Faire vérifier votre travail par le professeur.



a	b	c	S
I1	I2	I3	Q1
0	0	0	
0	0	1	
0	1	1	
0	1	0	
1	1	0	
1	1	1	
1	0	1	
1	0	0	

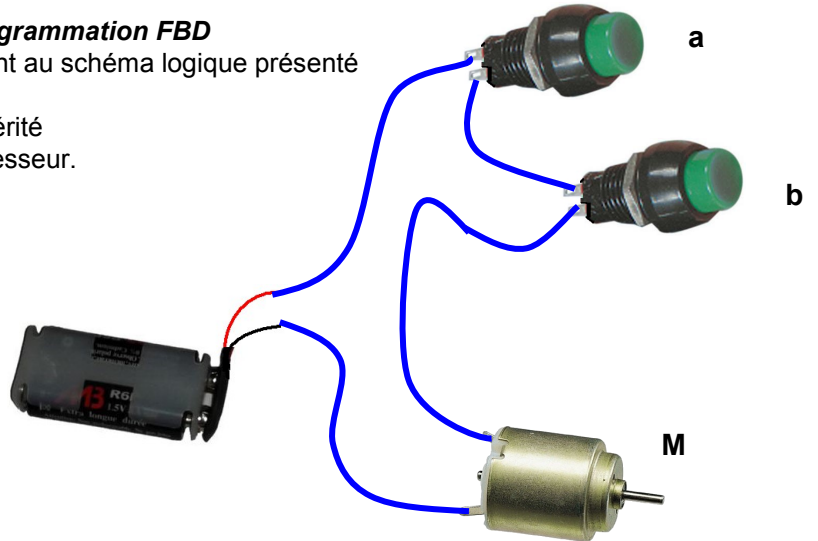
Deuxième partie : Mettre en œuvre le logiciel ZELIOSOFT - Langage FBD

2.1 Regarder la **Vidéo 03 - ZELIO - Configuration FBD** et configurer le logiciel ZELIOSOFT comme présenté.

2.2 Soit le circuit électrique ci-contre présenté dans la première partie.

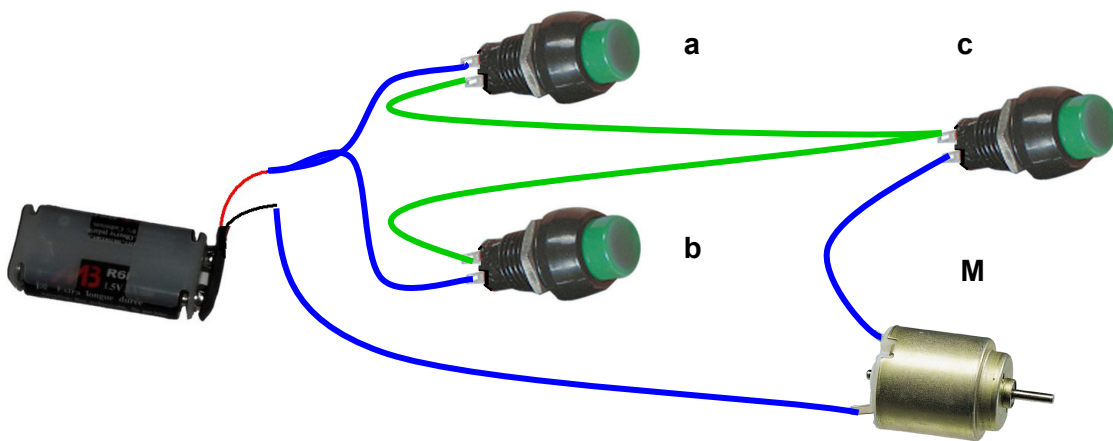
- Regarder la **Vidéo 04 - ZELIO - Programmation FBD**
- Réaliser le programme correspondant au schéma logique présenté dans la vidéo
- Vérifier les résultats de la table de vérité
- Faire vérifier votre travail par le professeur.

a	b	M
0	0	0
0	1	0
1	1	1
1	0	0



2.3 Soit le circuit électrique ci-dessous.

- Traduire le schéma électrique en schéma logique
- Réaliser le programme en langage FBD.
- Compléter la table de vérité. Comparer avec la table obtenue dans la première partie de l'activité
- Faire vérifier votre travail par le professeur.

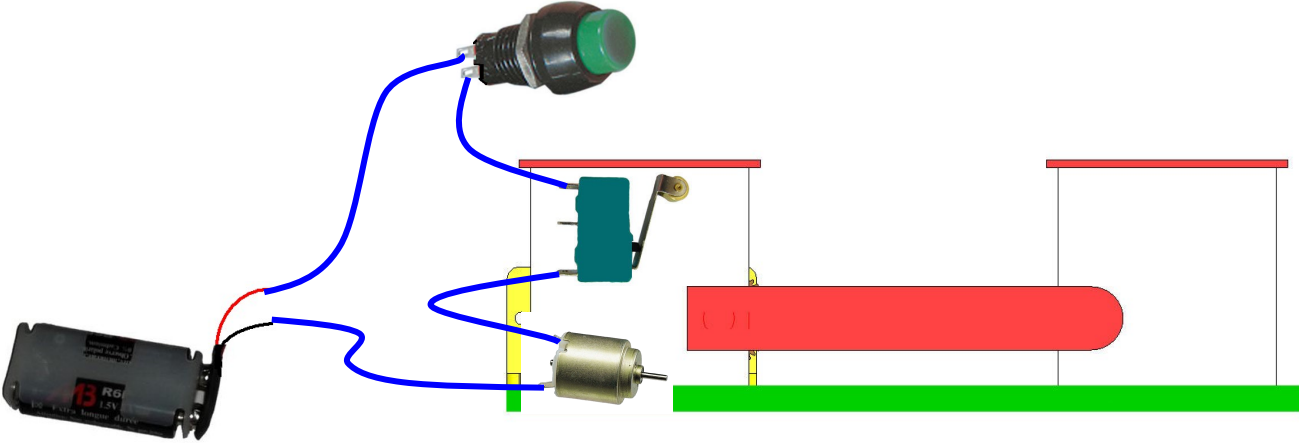


a	b	c	S
I1	I2	I3	Q1
0	0	0	
0	0	1	
0	1	1	
0	1	0	
1	1	0	
1	1	1	
1	0	1	
1	0	0	

Troisième partie : Réaliser et tester un programme en situation réelle

Le circuit électrique ci-dessous propose la solution permettant de lever la barrière jusqu'au contact sur le détecteur de fin de course.

On souhaite réaliser la même action mais par programmation sur automate ZELIO



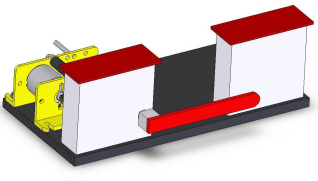





3.1 Traduire le schéma électrique en schéma logique



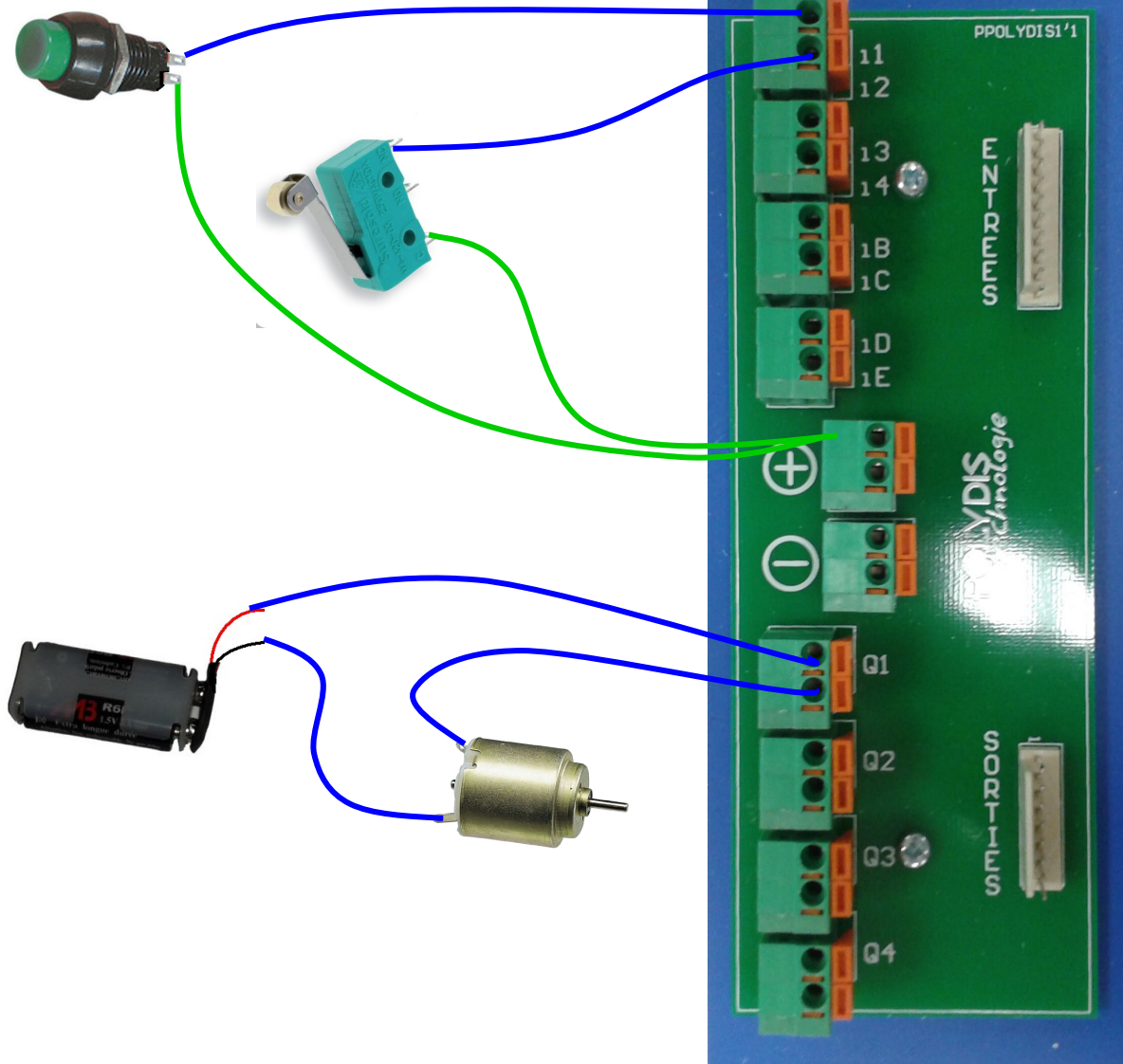
3.2 Réaliser le programme en langage FBD en utilisant les variables suivantes :

Eléments	Variables
Bouton poussoir	I1
Fin de course	I2
Moteur	Q1

3.3 En possession du matériel listé ci-dessous :

		
Maquette de Barrière	Alimentation - Support de piles LR6	Bouton poussoir
		
Platine Automate	Fil souple	Capteur fin de course à galet

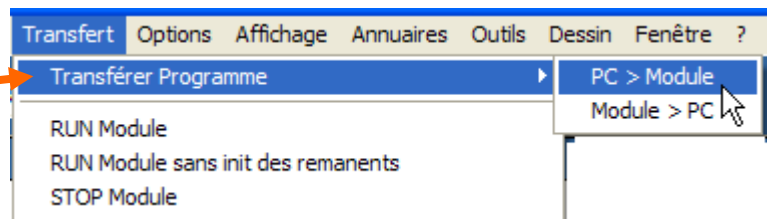
Réaliser le câblage des composants sur la platine.



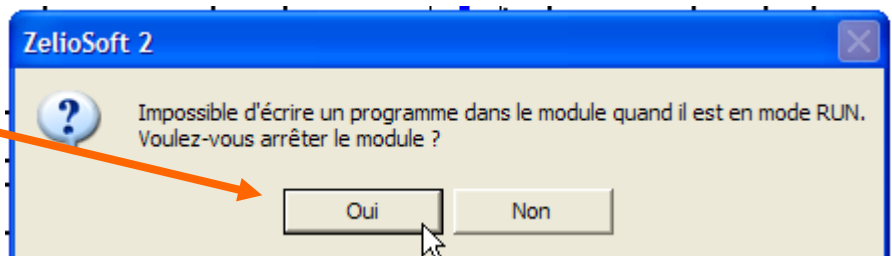
3.2 L'ensemble étant relié, nous allons transférer le programme dans l'automate, démarrer le programme (Run) depuis l'ordinateur et visualiser le fonctionnement de la maquette.

Pour cela :

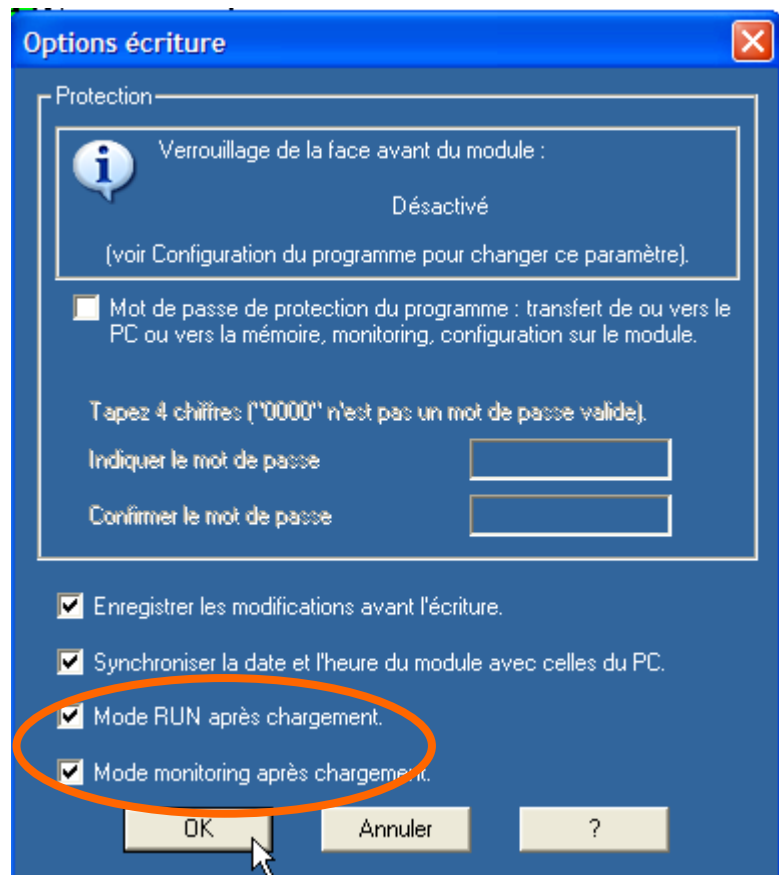
- Transférer le programme du PC > Module



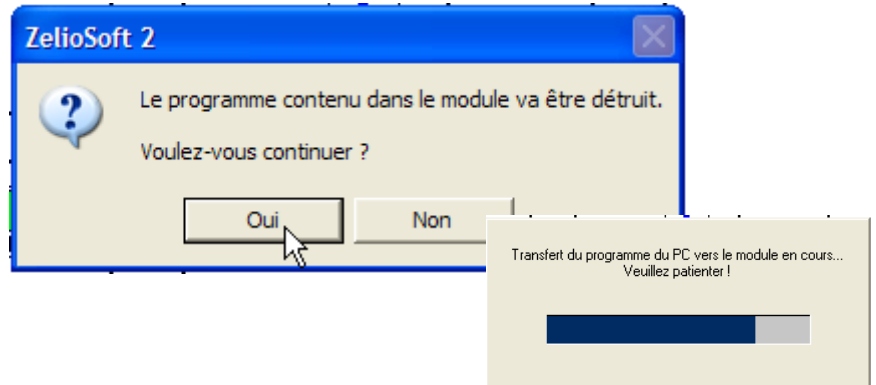
- Si le programme est en cours de fonctionnement dans l'automate, cliquer sur Oui



- Cocher les options, puis valider par **OK**



- Cliquer sur **Oui** pour valider la destruction du programme contenu dans l'automate. Le transfert démarre



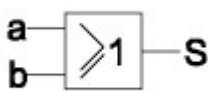
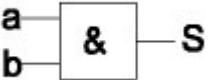


3.3 Actionner les capteurs et détecteurs. Constaté l'évolution de l'état des entrées et des sorties sur l'écran. Vérifier le bon fonctionnement de l'ensemble.

Fonctions logiques

Pour définir chacune des fonctions logiques, il existe plusieurs représentations :

- une représentation graphique : symbole logique
- une représentation algébrique : équation
- une représentation arithmétique : table de vérité

Nom de la fonction	Symbole	Equation	Table de vérité															
OUI		$S = a$	<table border="1"> <thead> <tr> <th>a</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	S	0	0	1	1									
a	S																	
0	0																	
1	1																	
NON		$S = \bar{a}$	<table border="1"> <thead> <tr> <th>a</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	a	S	0	1	1	0									
a	S																	
0	1																	
1	0																	
OU		$S = a + b$	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	a	b	S	0	0	0	0	1	1	1	1	1	1	0	1
a	b	S																
0	0	0																
0	1	1																
1	1	1																
1	0	1																
ET		$S = a \cdot b$	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	a	b	S	0	0	0	0	1	0	1	1	1	1	0	0
a	b	S																
0	0	0																
0	1	0																
1	1	1																
1	0	0																