

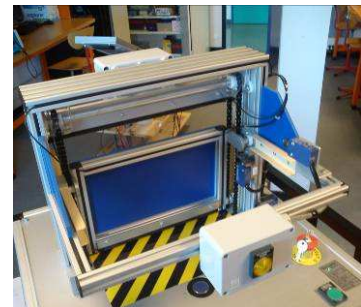
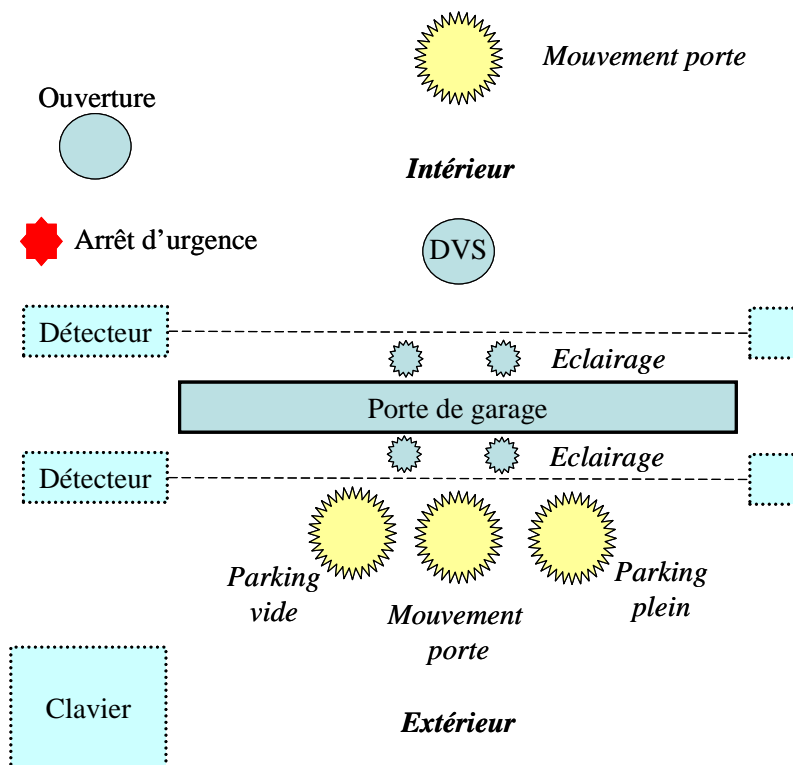
Automatisation d'une porte de garage

L'objectif de cette séance est d'automatiser le fonctionnement d'une porte de garage en utilisant un automate ALLEN BRADLEY que l'on programmera avec le logiciel RSLogix 5000.

1. Description

La maquette de la porte de garage automatisée sécurisée comprend :

- Une armoire électrique constituée de :
 - un interrupteur, permettant la mise en/ou hors tension de l'ensemble de l'armoire,
 - un bouton d'arrêt d'urgence, permettant d'interrompre l'alimentation du moteur,
 - un commutateur auto/manu,
 - un voyant de mise sous tension.

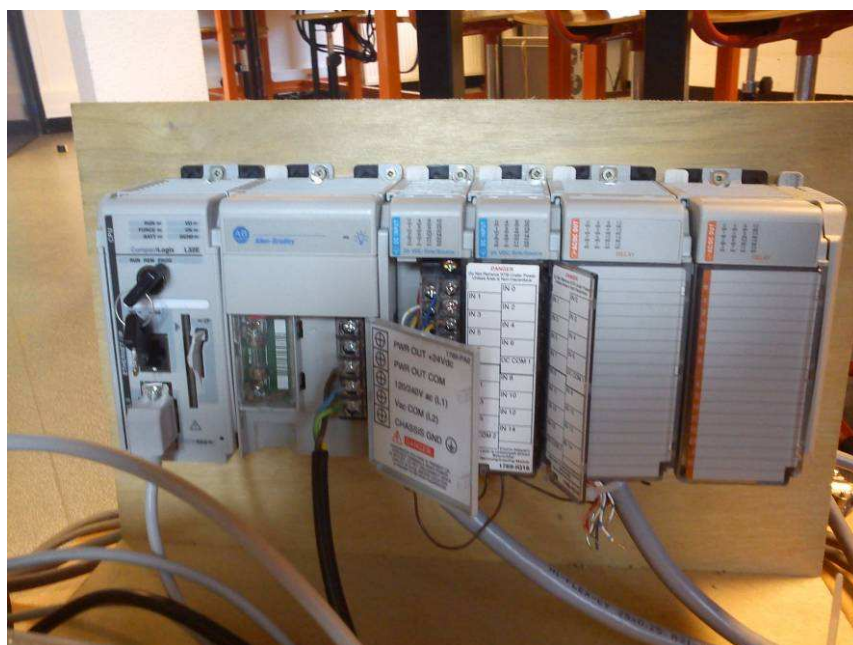


- Une partie opérative sur laquelle sont implantés :
 - 2 capteurs photoélectriques et fibres optiques (SECO et SECF) situés en dessous et au dessus de la porte afin de détecter un éventuel écrasement,
 - un capteur inductif (DVS) dans le sol à l'intérieur du parking, permettant la détection d'une voiture voulant quitter ce dernier,
 - un bouton "Commande Ouverture" (COM) qui provoque l'ouverture de la porte en cas d'appui par un utilisateur non motorisé situé à l'intérieur du parking,
 - 2 capteurs photoélectriques (système barrage avec réflecteurs : CBS et CBE) qui sont situés de chaque coté de la porte et permettant de détecter le passage de la voiture,
 - 2 interrupteurs mécaniques de fin de course des mouvements d'ouverture et de fermeture de la porte (FCO et FCF),

- un clavier (non utilisé lors de cette séance) qui permet à un utilisateur extérieur de rentrer un code pour ouvrir la porte,
- un moteur avec réducteur de vitesse et limiteur de couple mécanique permettant les mouvements de la porte,
- 4 voyants blancs, 2 situés à l'extérieur du garage et 2 à l'intérieur simulant l'éclairage de la zone d'accès sécurisée,
- 2 voyants "Parking VIDE" et "Parking PLEIN" situés au-dessus de l'entrée à l'extérieur du parking indiquant l'état du parking à l'utilisateur : le voyant vert restant allumé tant que le parking n'est pas plein, et le voyant rouge pour signaler que le parking est plein
- 2 voyants jaunes "mouvement porte", dont l'un est à l'intérieur et l'autre à l'extérieur, s'allumant pour indiquer que la porte est en mouvement.

2. Branchement des entrées et des sorties

Les connexions avec tous les capteurs et actionneurs de la maquette décrite ci-dessus, sont regroupées sur deux boîtiers (un boîtier d'entrées et un boîtier de sorties). L'automate mis à disposition comporte deux cartes à 16 entrées T.O.R. (emplacement 1 et 2) et deux cartes à 16 sorties T.O.R. (emplacement 3 et 4, cette dernière carte ne sera pas utilisée dans ce TP). Tout le câblage de l'automate a été préalablement réalisé.



Compléter le tableau suivant (à rendre dans votre compte rendu) en précisant l'adressage des entrées et si les contacts sont normalement ouverts (NO) ou normalement fermés (NF).

adresse	description	mnémonique	NO / NF
	Arrêt d'urgence sur la maquette	ARU	
%I:1/7	Porte ouverte	FCO	NO
	Cellule barrages (intérieur parking)	CBS	
%I:1/5	Sécurité porte à l'ouverture	SECO	NF
	Présence véhicule sur capteur	DVS	
	Cellule barrages (extérieur parking)	CBE	
%I:1/1	Sécurité porte à la fermeture	SECF	NF
	Commutateur Auto/Manu sur l'armoire	MANU	
	Bouton poussoir "commande ouverture porte"	COM	
%I:1/6	Porte fermée	FCF	NO

Les sorties utilisées dans cette séance de travaux pratiques sont :

adresse	description	mnémonique
%O:3/2	Voyant parking vide	VIDE
%O:3/3	Ouverture porte + Voyant mouvement porte	OUV
%O:3/4	Eclairage parking	ECL
%O:3/5	Voyant parking plein	PLEIN
%O:3/6	Fermeture porte + Voyant mouvement porte	FERM

3. Prise en main du logiciel de programmation : RSLogix 5000 French

Connectez vous sur « *cet ordinateur* » en mode local (utilisateur *usrlocal* ; mot de passe *usrlocal*).

En aucun cas vous ne sauvegarderez de fichiers sur les ordinateurs de la salle de TP mais vous utiliserez clé USB ou votre compte.

La première étape consiste à ce que votre ordinateur reconnaisse sur le réseau l'adresse de l'automate.

Lancer le logiciel « *RSLinx classic* ». Dans le menu « *communication* » ; « *configure drivers* » ; « *available driver types* » ouvrir le menu déroulant et choisir « *ethernet / IP Driver* ».

« *add new* » ; « *ok* » puis cliquer sur « *browse remote subset* » .

Renseigner l'adresse IP qui commence par 172 ; les derniers chiffres étant indiqués sur votre PC (par exemple 172.20.81.99).

Pour le « *subnet mask* » renseigner : 255.255.0.0

Cliquer sur « *ok* » ; « *close* » et quitter « *RSLinx classic* » avec la commande « *exit* » du menu « *file* ».

La seconde étape consiste à utiliser le logiciel *RSLogix 5000 French* (sur le bureau).

Dans « *fichier* » ; « *nouveau* » ; choisir dans le menu déroulant le type de l'automate : il s'agit d'un 1769 L32 E (indiqué sur le boîtier de l'automate). Donner un nom à votre programme puis « *ok* ».

Pour décrire la configuration matérielle de votre automate : clic droit sur « *compact bus local* » ; « *nouveau module* » et ajouter des modules numériques (car nous disposons de 2 cartes d'entrées en Tout Ou Rien et d'une carte de sorties en Tout Ou Rien. En vérifiant leurs emplacements dans l'automate, choisir les types suivants : 2 cartes d'entrées 1769IQ16 et 1 carte de sorties 1769OW16.

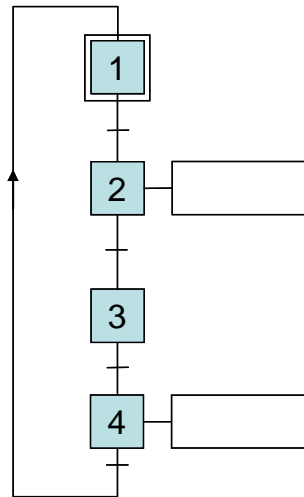
Vous pouvez associer le nom de chaque entrée à une étiquette qui la décrit. Pour ce faire, dans « *tags du controller* » ; clic droit puis « *éditer les tags* » ; visualisez les données (data) des cartes d'entrées (input : I) et de sorties (output : O) en cliquant sur les « + » pour ouvrir les data associés aux input et output. On peut alors dans la colonne « *description* » associer les mnémoniques (étiquettes) aux data en tapant leurs noms. Créer les mnémoniques pour les entrées et les sorties : FCO, CBS, DVS, CBE, SECF, MANU, COM, FCF, OUV, ECL et FERM.

Afin d'associer des étapes à des bits internes pour décrire des graficets par des structures ladder, il est nécessaire de déclarer ces bits internes. Ceux ci sont définis comme des « *tags du program* » dans « *main program* ». Clic droit puis « *éditer les tags* ». On associe un « *nom* » (étape 17 par exemple) comme étant de type booléen « *data type : bool* ». On procède ainsi pour tout bit interne dont on a besoin dans le programme (en particulier pour chacune des étapes).

4. Cahier des charges : mode manuel

Lorsque le commutateur « *Auto/Manu* » sur l'armoire est placé en position Manu, alors l'appui sur le bouton poussoir « *commande ouverture porte* », doit provoquer l'ouverture de la porte jusqu'à ce que le capteur « *porte ouverte* » s'enclenche. Puis, lors d'un nouvel appui sur « *commande ouverture porte* » la porte doit se refermer entièrement jusqu'à ce que le capteur « *porte fermée* » s'enclenche.

Compléter sur votre compte rendu le grafcet suivant qui décrit ce cahier des charges.

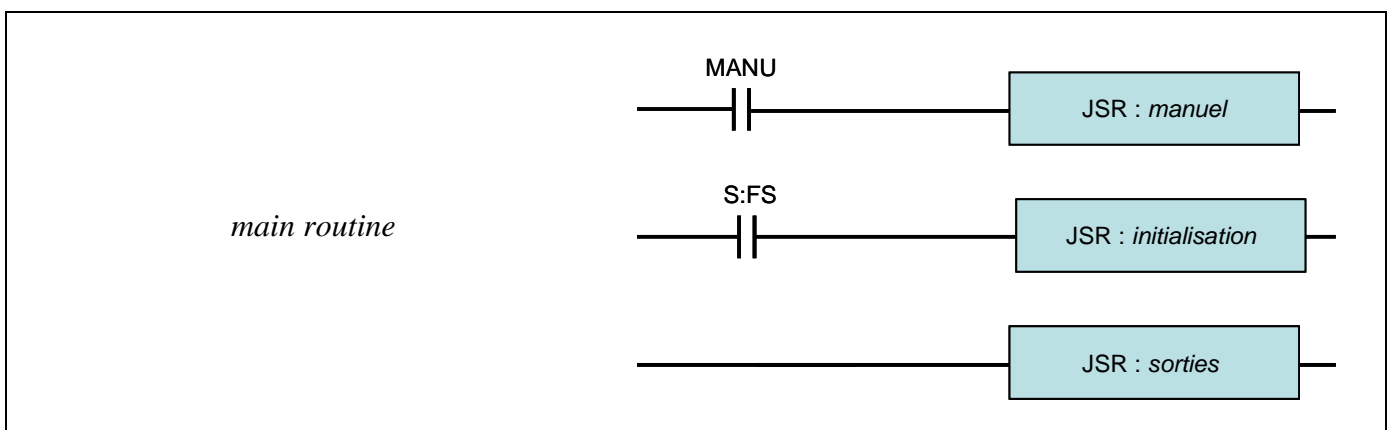


Créer les bits internes *etape_1* ; *etape_2* ; *etape_3* ; *etape_4* comme des « *tags du program* ».

Pour programmer l'automate conformément à ce cahier des charges, vous allez créer les trois structures ci dessous en plus du programme principal « *main routine* ». Pour ce faire clic droit sur « *Mainprogram* » pour créer une « *nouvelle routine* ». En double cliquant sur une routine, celle ci s'ouvre pour être éditée. Vous avez alors accès à différents symboles (relais, bobines, blocs, ...) proposés dans de nombreux onglets. N'hésitez pas à les faire défiler horizontalement si nécessaire). En cas de besoin, une aide est disponible en tapant sur « *F1* ».

- la routine *initialisation*
Celle ci sera scrutée au démarrage de l'automate et devra donc consister à l'activation de l'étape initiale (load *etape_1*) et à la désactivation des autres étapes.
- La routine *manuel*
Celle ci décrit la structure du grafcet, c'est à dire comment on passe d'une étape à l'autre. Par exemple si l'*etape_1* est active et que COM est vrai alors l'*etape_2* est activée et l'*etape_1* est désactivée.
- La routine *sorties*
Celle ci associe chaque sortie à ses conditions d'activation (activation des étapes). Par exemple l'action "OUV" est associée à l'*etape_2*. Lire attentivement l'annexe en dernière page.

Le programme principal (*main routine*) contient les instructions d'appel des autres routines (*JSR* dans l'onglet *contrôle du programme*). Par exemple, la routine *initialisation* ne doit être appelée que lors du démarrage (premier cycle de l'automate). Le bit interne correspondant au premier cycle de l'automate est **S:FS** . La routine *sorties* est scrutée à chaque cycle de l'automate. La routine *manuel* est scrutée à chaque cycle de l'automate tant que « *MANU* » est vrai.



Avant de charger l'ensemble des routines dans l'automate, une vérification est possible en cliquant sur l'icone « *vérifier la routine* » et les informations s'affichent alors en bas de l'écran.

Une fois qu'il n'y a plus d'erreurs, le programme peut être chargé dans l'automate. La clé sur l'automate doit être en position « *Rem* ». Pour ce faire, il est nécessaire de définir le chemin. Pour ce faire, cliquer sur le bouton « *qui actif ...* », cliquer sur « *AB_ETHIP-1, Ethernet* » et choisir le « *00, compactLogix processor* » puis cliquer sur « *charger* ».

Pour faire des modifications, rebasculer en mode « *hors ligne* ».

Vérifier avec l'enseignant que le fonctionnement de la maquette respecte bien le cahier des charges.

5. Cahier des charges : mode automatique – sortie de voiture

Compléter le programme précédent de manière à décrire le cahier des charges suivant correspondant à l'ajout du mode automatique (en plus du mode manuel). On ajoutera une nouvelle routine.

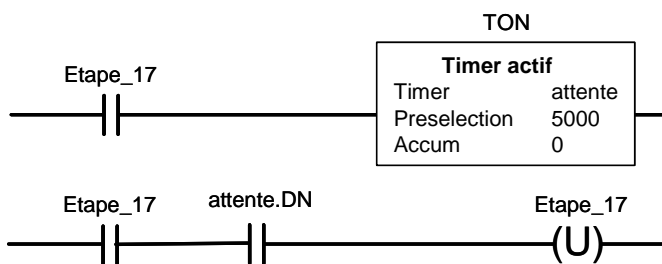
Lorsque le commutateur "Auto/Manu" sur l'armoire est placé en position "Auto", alors :

- si une voiture arrive sur le capteur DVS, la lumière s'allume, la porte s'ouvre totalement. La porte reste alors ouverte pendant 2 secondes ; pendant ce temps la voiture est immobilisée. Puis passé ces deux secondes, et une fois que le véhicule est sorti, la porte se referme, puis la lumière s'éteint.

Pour utiliser une fonction qui compte le temps, vous devez définir dans les « *tags du program* » une temporisation que vous appellerez « *attente* » ; son « *data type* » étant « *timer* ».

Ensuite vous trouverez dans le menu « *temporisateur compteur* » une temporisation de type TON qui fonctionne comme illustré ci contre.

A l'activation de l'Etape_17, la temporisation commence. Au bout de 5000 ms, attente.DN devient vrai et Etape_17 est mis à 0. Ainsi la durée d'activation de Etape_17 est de 5s.



6. Cahier des charges : mode automatique – sortie de piéton

Compléter le programme "mode automatique" précédent de manière à décrire en plus le cahier des charges suivant.

- Si à l'intérieur du parking un piéton veut sortir, l'appui sur le bouton poussoir "commande ouverture porte", doit provoquer l'ouverture de la porte jusqu'à ce que le capteur "porte ouverte" s'enclenche. La porte reste alors ouverte pendant 5 secondes. Puis la porte doit se refermer entièrement jusqu'à ce que le capteur "porte fermée" s'enclenche. Durant ce mode « *sortie de piéton* », la présence d'une voiture sur DVS est sans effet jusqu'à la refermeture complète de la porte. Pendant toute cette procédure, l'éclairage doit être activé.

7. Cahier des charges : mode automatique – sécurité écrasement

Compléter le programme "mode automatique" précédent de manière à décrire en plus le cahier des charges suivant.

- Si en se refermant, la porte rencontre un obstacle, elle doit s'arrêter pendant une seconde, puis s'ouvrir complètement et attendre 1 seconde avant d'essayer à nouveau de se fermer.

8. Cahier des charges : mode automatique – utilisation d'un code

Compléter le programme "mode automatique" précédent de manière à décrire en plus le cahier des charges suivant.

- Si à l'extérieur, un utilisateur tape le bon code (que vous aurez préalablement défini), le voyant « code bon » s'éclaire (pendant 5 secondes), la porte s'ouvre comme en mode « automatique – sortie de piéton ». Si le code entré est erroné, le voyant « code mauvais » s'éclaire et l'éclairage doit clignoter pendant 5 secondes.

Proposer sur votre compte rendu les graficets permettant de décrire le cahier des charges du mode automatique.

Programmer l'API (Automate Programmable Industriel) afin qu'il réponde à ce cahier des charges.

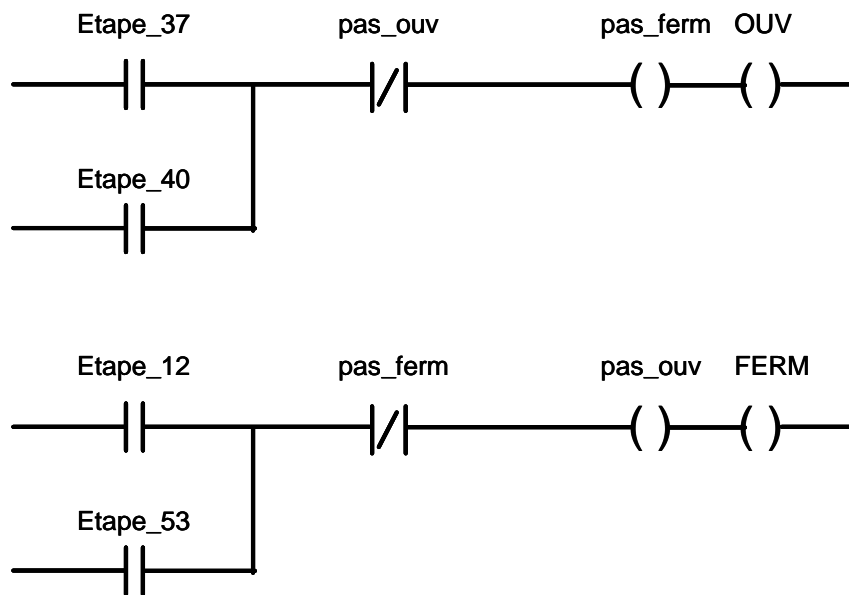
Effectuer les tests.

ANNEXE : consignes essentiels pour l'activation des sorties

Afin de ne pas détériorer la maquette, vous devez impérativement éviter d'activer en même temps les sorties OUV et FERM.

Pour ce faire, ajouter les bits internes pas_ferm et pas_ouv dans les « tags du program ».

Dans votre routine *sorties*, si vous désirez activer la sortie OUV aux étapes 37 et 40 et la sortie FERM aux étapes 12 et 53, respectez l'écriture suivante :



Il est essentiel que vous ayez compris cette annexe (dans le cas contraire, appeler l'enseignant) et que vous respectiez cette exigence de programmation.