Subrini Guillaume Taverne Arnaud Kabbaj Ismail

# Projet tuteuré

## Projet de prise en main des automates Omron

# Sommaire

1. Matériel
2. Cahier des charges
3. Câblage de la Maquette
4. Interface Homme machine
5. Programmation
6. Mode automatique
7. Mode manuel
8. Commande API
9. Commande Variateur de Fréquence

# Matériel

Nous disposons du matériel suivant :

* [Automate Omron CP1L](https://benl.rs-online.com/web/p/products/8211775/?grossPrice=Y&cm_mmc=FR-PLA-DS3A-_-google-_-PLA_FR_FR_CATCHALL-_-Catch+All+Ad+Group-_-PRODUCT_GROUP&matchtype=&pla-293946777986&gclid=Cj0KCQjwzunmBRDsARIsAGrt4mudXzzdF2T_D3vb-nD7RbfYerC9MYpg7Wf-Xc5_tbDCTU_ljgRLYOMaAqFxEALw_wcB&gclsrc=aw.ds) / <https://docs-emea.rs-online.com/webdocs/157d/0900766b8157db99.pdf>
* Variateur de fréquence Omron : <http://variateur-frequence.com/Omron/documentation/Quick%20guide%20MX2.pdf>
* Disjoncteur de protection du Driver
* Moteur : <https://www.mvi-sa.fr/moteur-electrique-mta-100-l6-4-0-65-2-2-4-6-art_fr_268284.html>
* Switch Ethernet S101-T : <https://www.materielelectrique.com/switch-ethernet-ports-p-104269.html>
* Boite de commutateur
* Capteur Inductif : <https://fr.rs-online.com/web/p/capteurs-inductifs/4474045/>
* IHM 320\*234 : <https://fr.rs-online.com/web/p/ecrans-tactiles-interface-homme-machine-ihm-/8211794/>

# Cahier des charges

Le but du projet est d’utiliser la maquette présentée afin de réaliser une chaine automatisée, ce qui nécessite :

* Préparation mécanique de la maquette
* Dessin et préparation IHM via NB-Designer
* Programmation API via CX-Programmer

Ces trois aspects devront être maitrisés par tous les membres du groupe dans une présentation finale.

### **Fonction :**

* Gestion d’un ascenseur de trois étages
* Calibration : on fixe la hauteur de chaque étage, vitesse moteur, accélération et décélération moteur.
* Verrouillage des paramètres et changement de mode par code pavé numérique.
* Mode automatique avec bouton d’appel et Capteur de présence pour chaque étage

(+ bouton position de sécurité pour Maintenance).

* Mode Manuel (Visualisation Montée/Descente et position et barre de gestion de la vitesse + commande asservissement de la position).
* Interface Homme/Machine pour chaque Mode (commande et visualisation) image bouton BMP
* Programmation d’une séquence d’action
* Fonction set -> 0 : Fixation de la position actuelle en tant que position 0m.
* Onglet Historique de commande récupérable sur clé USB (Maintenance).

# Cablage de la maquette :

Selon les valeurs de nos commutateurs, du capteur inductif ou de l’IHM nous allons commander le variateur de fréquence pour qu’il alimente le moteur comme nous le souhaitons.

Tous les systèmes de commande (IHM, Driver, API) sont reliés entre eux en Ethernet via le Switch.

Nous allons pouvoir gérer le sens de rotation et la vitesse du moteur afin de réaliser les fonctions du cahier des charges.

Nous avons rajouté des voyants pour visualiser l’état du moteur qu’elle que soit le mode de fonctionnement ainsi qu’un bouton d’arrêt d’urgence en entrée d’alimentation 230V de la maquette.

Nous utilisons 3 cellule inductive qui vous nous servir de détecteur de présence pour chaque étage.

# Interface Homme machine

L’interface Homme machine est un écran qui nous permet de commander directement le programme ou de réaliser une visualisation de l’état du programme.

Réf : <https://fr.rs-online.com/web/p/ecrans-tactiles-interface-homme-machine-ihm-/8211794/>

Ce type d’écran ne peut être programmer par CX-designer nous avons donc décider de la programmer avec NB-designer.

Nous avons réalisé chaque icone à part pour ensuite les intégrer dans nos différentes pages de visualisation.

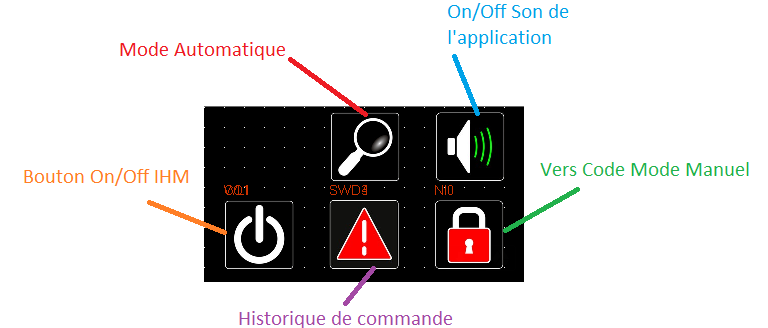
Chaque composant est relié à une variable du programme CX-programmer ce qui nous permet de la commander ou de visualiser la valeur de cette variable.

De plus nous avons réaliser un blocage de L’IHM grâce à une entrée sur l’API qui nous permet de créer une sécurité supplémentaire (Cf. table des variables).

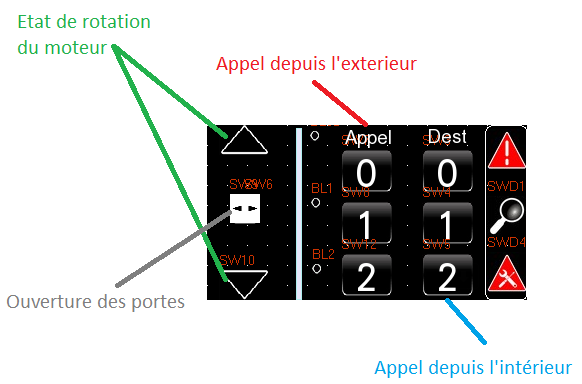
Nous allons exposer les écrans les plus importants tel que :

* Menu principal
* Mode Automatique
* Mode Manuel

## Menu principal :

Si le commutateur de sécurité est désactivé nous accéderons d’abord à cet page qui nous permet de naviguer entre tous les mode du programme :

## Mode Automatique :

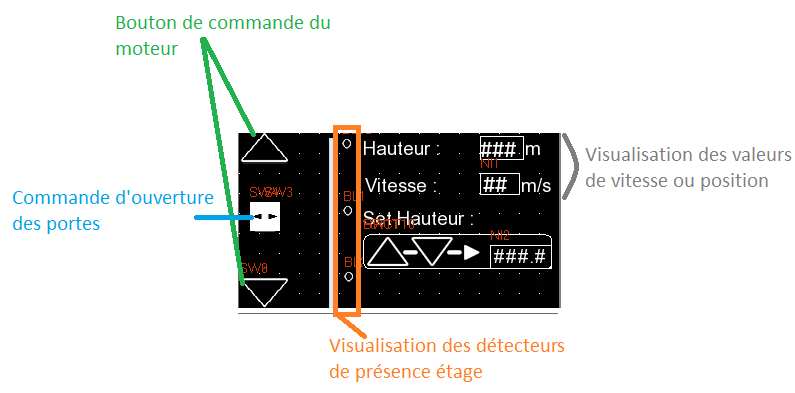
Dans ce mode nous utilisons le système avec les commandes physiques c’est-à-dire les boutons et les commutateurs afin que le système fonctionne en complète autonomie, nous pouvons visualiser l’état des différents actionneurs du programme comme les capteurs de position de chaque étage ou le bouton d’appel qui a été enclenché.

## Mode Manuel :

Pour accéder au mode manuel il faut d’abord taper un code de sécurité que nous choisissions dans le programme de l’API afin de sécuriser le système.



Ici le code pour accéder au mode Manuel est 2018.

Dans le mode manuel nous pouvons commander directement le moteur par appuie sur la commande de l’IHM ce mode est utilisable pour les phases de maintenance du système.

# Programmation

## Mode automatique

Ce mode permettra au système de fonctionner de manière autonome sans l’intervention d’un tiers. Il reprendra ainsi les principes de base d’un ascenseur avec l’appel à un étage la commande interne, l’ouverture et fermeture des portes.

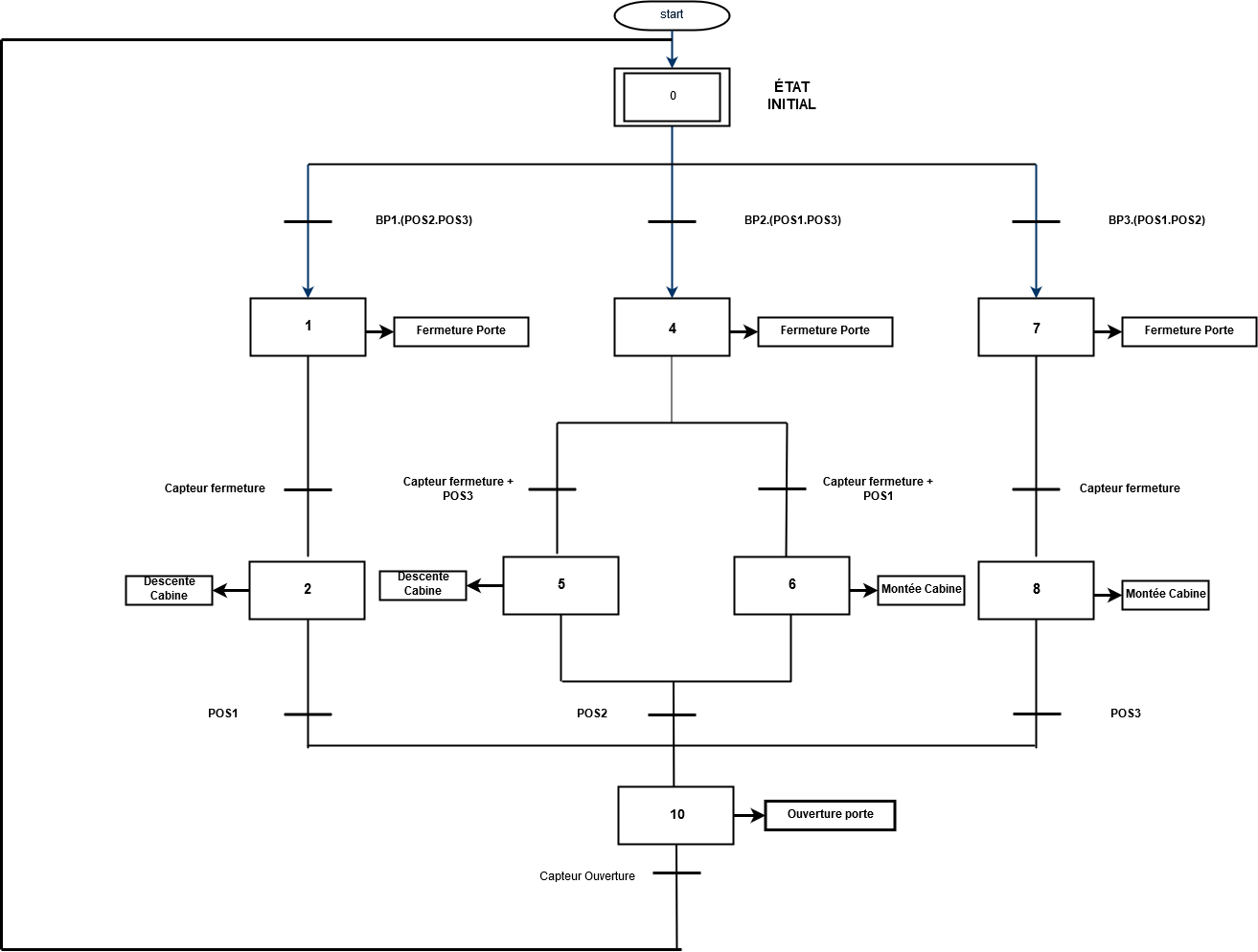
A travers l’IHM, l’utilisateur aura la possibilité de superviser le système grâce aux informations suivante :

* + Position de la cabine (capteurs à chaque étage : POS1, POS2, POS3)
  + Vitesse/accélération/décélération

## Table des variables :

* + BP1 : Appel étage 1
  + BP2 : Appel étage 2
  + BP3 : Appel étage 3
  + Pos1 : Capteur cabine à l’étage 1
  + Pos2 : Capteur cabine à l’étage 2
  + Pos3 : Capteur cabine à l’étage 3

## Grafcet mode Automatique

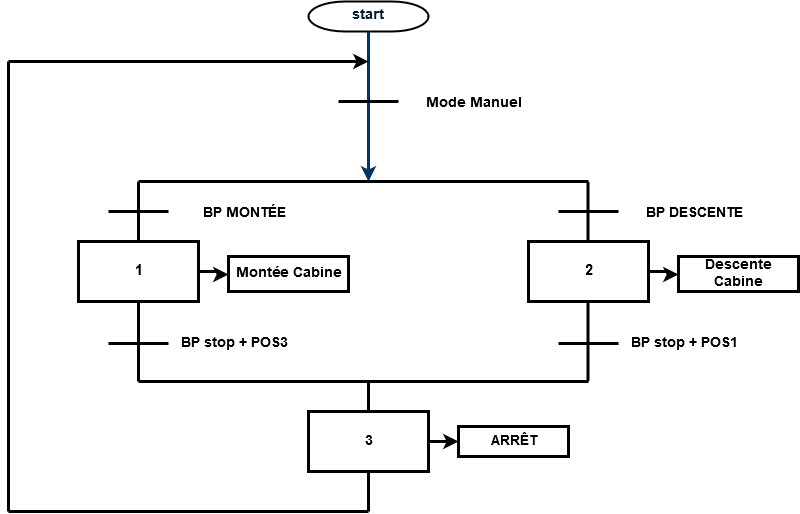


Grace à cette configuration nous n’avons pas besoin de détecter la position de la cabine au départ du programme.

## Mode manuel

Ce mode permettra à l’utilisateur de piloter manuellement le système, ce mode peut être utiliser en cas de maintenance du système et possède les fonctions suivantes :

* Réglage de la position exacte de la cabine
* Activer la montée /descente avec un système d’auto maintien
* Arrêter la cabine avec un bouton stop
* Nous pourrons programmer une séquence d’action à reproduire
* Tout le système sera supervisé par l’IHM en temps réel.

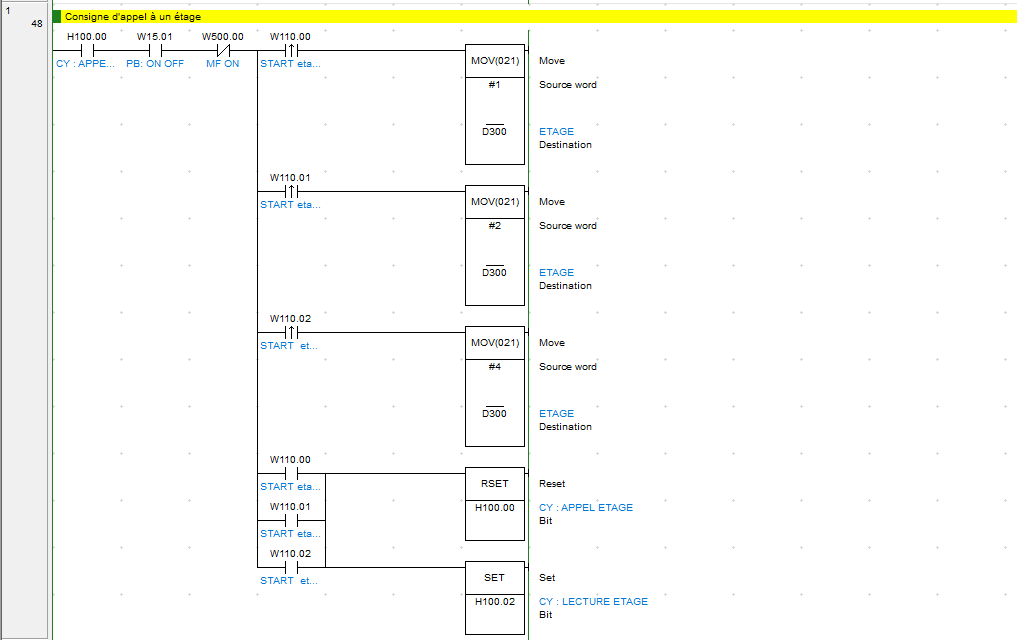


## Commande API

Nous allons commander notre système de cette façon :

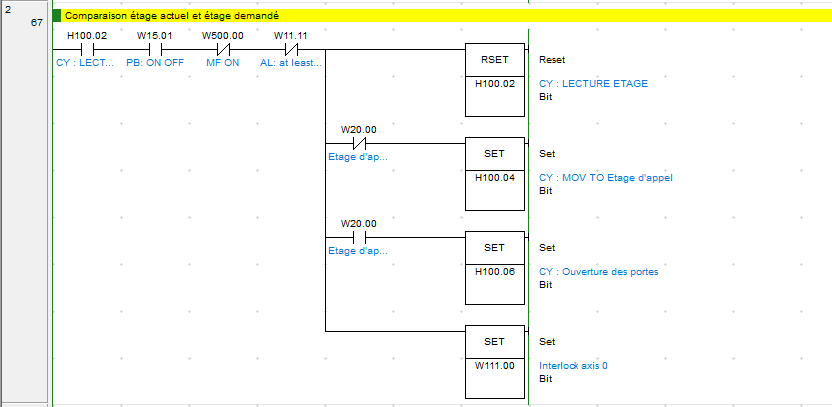
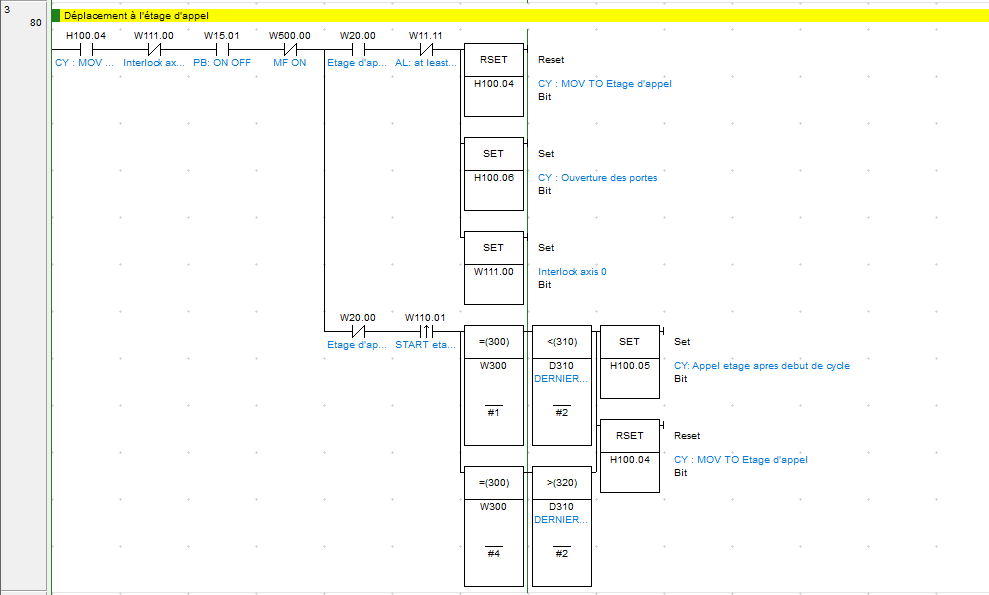
*Table des variables*

Le programme final est un cycle avec plusieurs étages qui se basent du Grafcet montrée précédemment.

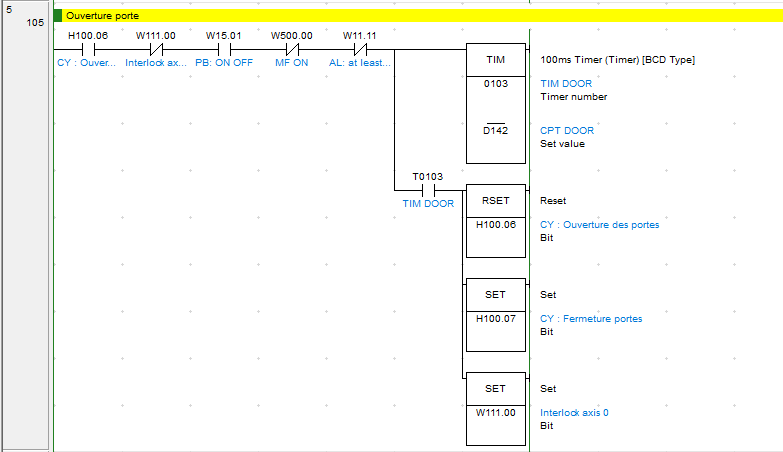
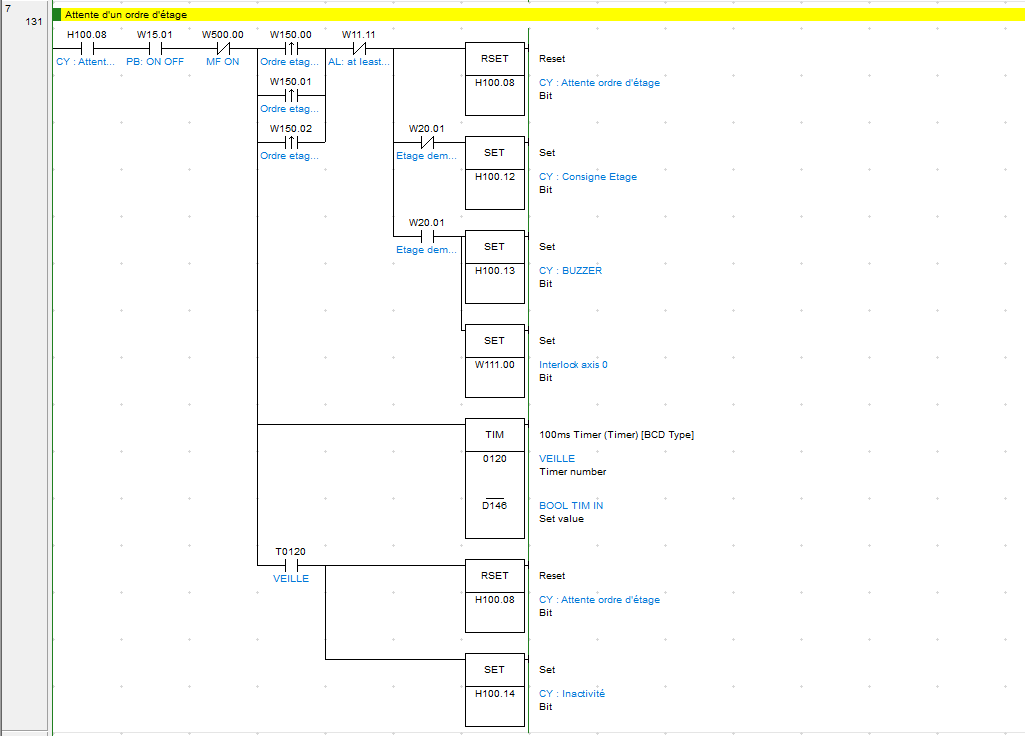
1. Nous commençons le programme lorsque que le système reçoit une consigne d’étage de destination :

Ici nous attendons la réception d’un appel externe c’est-à-dire que quelqu’un cherche à monter dans l’ascenseur, la commande de la destination se fais de la même façon mais avec des variables différentes. Ici nous utilisons les variables suivantes :

* W110.00 POUR ETAGE 0
* W110.01 POUR ETAGE 1
* W110.02 POUR ETAGE 2

1. Une fois qu’un appel à été effectué nous devons vérifier si la cabine est déjà à l’étage souhaitée et doit directement ouvrir les portes ou si elle doit se déplacé.
2. Cycle de déplacement

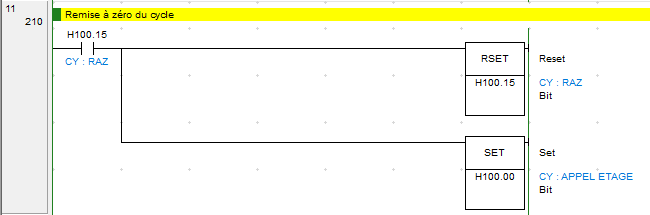
La partie basse du code vérifie si pendant le déplacement vers un étage on à un appel à un étage intermédiaire, mais cette fonction n’est pas prise en compte car risque de poser des problèmes dans le code (pas assez de temps pour finaliser cette partie).

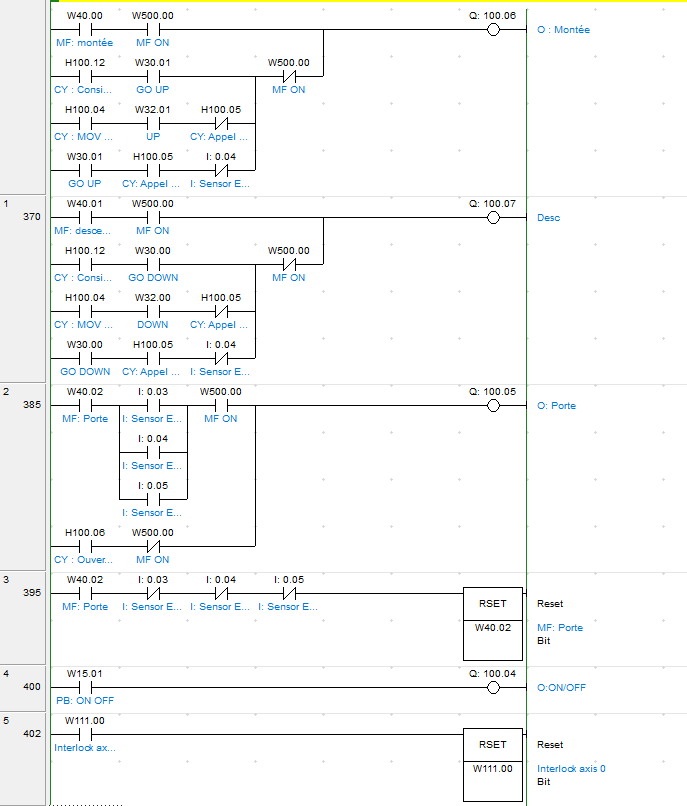
1. Une fois arrivé à destination nous ouvrons les portes et nous laissons le temps à l’utilisateur de montée, 10s avant de lancer à fermeture des portes.
2. Fermeture des portes qui dure également 10s

Et après cette étape s’il s’écoule plus de 20 secondes sans qu’aucune commande ne soit actionné le cycle se resetera.

Chaque Partie de ce cycle est reconnaissable avec un contact NO de type H100.XX.

Fonction de remise à zéro :

Cette fonction est appelée lorsque nous voulons remettre le programme à &ero est revenir à l’état initial.

Fonction de commande des sorties Montée Descente ou Porte :

Cette fonction est à part du reste du programme est tourne en permanence afin de commander les sortie MONTEE, DESCENTE et Ouverture de Porte quand il le faut, dans cette fonction on fait la distinction entre les appel externes et internes ainsi que le mode de fonctionnement Automatique ou Manuel.

Nous avons également une autre fonction qui nous permet de garder en mémoire le dernier étage atteint par la cabine.

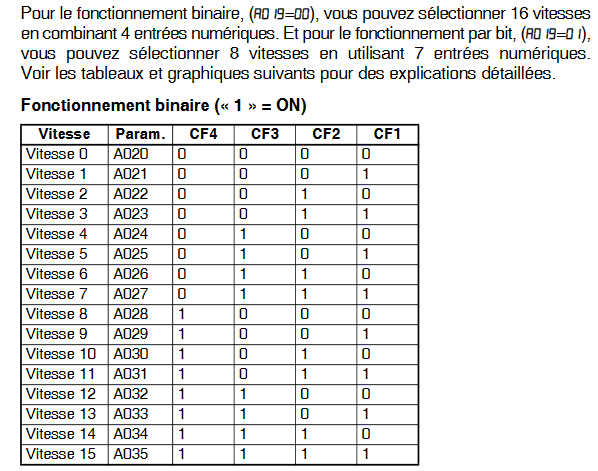
## Commande du Variateur de fréquence

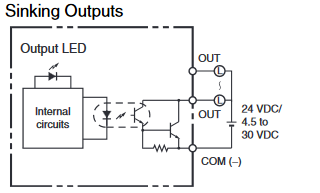
Pour commander le Variateur de fréquence nous avons deux solutions

* Utiliser un module de communication CP1W CIF11 pour commander le variateur par Ethernet
* Utiliser les pins de l’automate directement est programmer le variateur en parallèle afin qu’ils répondent à nos commandes de façon analogique ou binaires.

Nous avons décider de le commander de façon binaire , pour cela nous devons faire quelques réglages :

* + F001= fréquence de sortie qui sera 30Hz ici.
  + A001=02 commande en binaire
  + Nous utilisons 4 pins que nous allons commander par l’automate pour fixer la vitesse de rotation du moteur.



Nous nous sommes rendu compte que les sorties de l’automate fonctionnent par transistor et non par relais il nous faut donc brancher un pôle commun entre l’API et le variateur de fréquence pour que les sorties fonctionnement, c’est également pour cette raison que nous ne pouvons pas utiliser les voyants tel qu’ils sont branchés actuellement.

Nous avons donc raccordé le COM de l’API avec le P24 du variateur en tant que pôle commun et nous avons utilisé les entrées standards configurés tel que :

1 : Accélération Marche avant

2 : Accélération marche arrière

Ainsi nous pouvons commander directement l’accélération de notre moteur dans un sens ou dans l’autre.

# Conclusion :

Ce projet était divers et complet, il nous a fait découvrir les automates Omron ainsi que les Interface de développement qui lui sont lié.

A cause de problèmes de concordance entre certains composants nous n’avons pas pu mettre en place toutes les fonctionnalités que nous voulions dans notre cahier des charges mais nous obtenons un système fonctionnel.

Il y a plusieurs perspectives d’amélioration tel qu’une page de paramètres en début de programme afin de spécifier les caractéristiques mécaniques du système comme la puissance du moteur commandé ou le taux du réducteur auquel le moteur est relié qui sont des paramètres essentiels à un véritable ascenseur.

En utilisant un Drive pour commander les moteurs au lieu d’un variateur nous aurions également pu développer la commande du moteur de façon plus précise et récupérer des valeurs de vitesse et de position.