

# TRAVAUX PRATIQUES PROJET AUTOMATISME 4A

SUJET:

**GESTION D'UN ASCENSEUR**


TUTEUR:

BRISSON ALARIC 

GROUPE 4GSI - TD2 - TP4:

NGUYEN Huynh Dung 

LE Quang Quoc Bao 

NGUYEN Thi Thuy Dung 

MADOUYEME Richard 

# Sommaire

## INTRODUCTION

I) Présentation

II) Les schémas

III) Introduction du logiciel

IV) Programmation du grafcet

## CONCLUSION

## **INTRODUCTION :**

L'évolution de la technologie a fait de l'ascenseur un outil plus qu'important et très courtisé. Grace à son application, plus de souci pour qui prospecte construire des bâtiments de plusieurs étages ; dans la médecine, le transport des malades d'un niveau à un autre, dans la vie courante, le déplacement des handicapés ou encore le gain de temps pour les personnes ayant toutes leurs facultés physiques sont sans failles.

Ingénieurs de demain, nous nous intéressons à des procédés plus pratiques et faciles pour l'Homme, ce qui explique notre choix pour ce projet.

## I. Présentation:

Ce projet est réalisé par un groupe de 4 étudiants et chaque étudiant, chaque étudiant ayant un rôle bien défini dans la réalisation du projet (gestion des tâches par diagramme de Gantt). Il a une durée totale de 20 heures réparties de la manière : 1h20 en TD : explication des projets, constitution des groupes, début de planification et 5 séances de 4h en TP : Réalisation du projet.

Pour y arriver, nous nous sommes servi du logiciel Unilogic : Simple, intuitif, environnement de développement et utilitaires riches en fonctionnalités.

### But de TP:

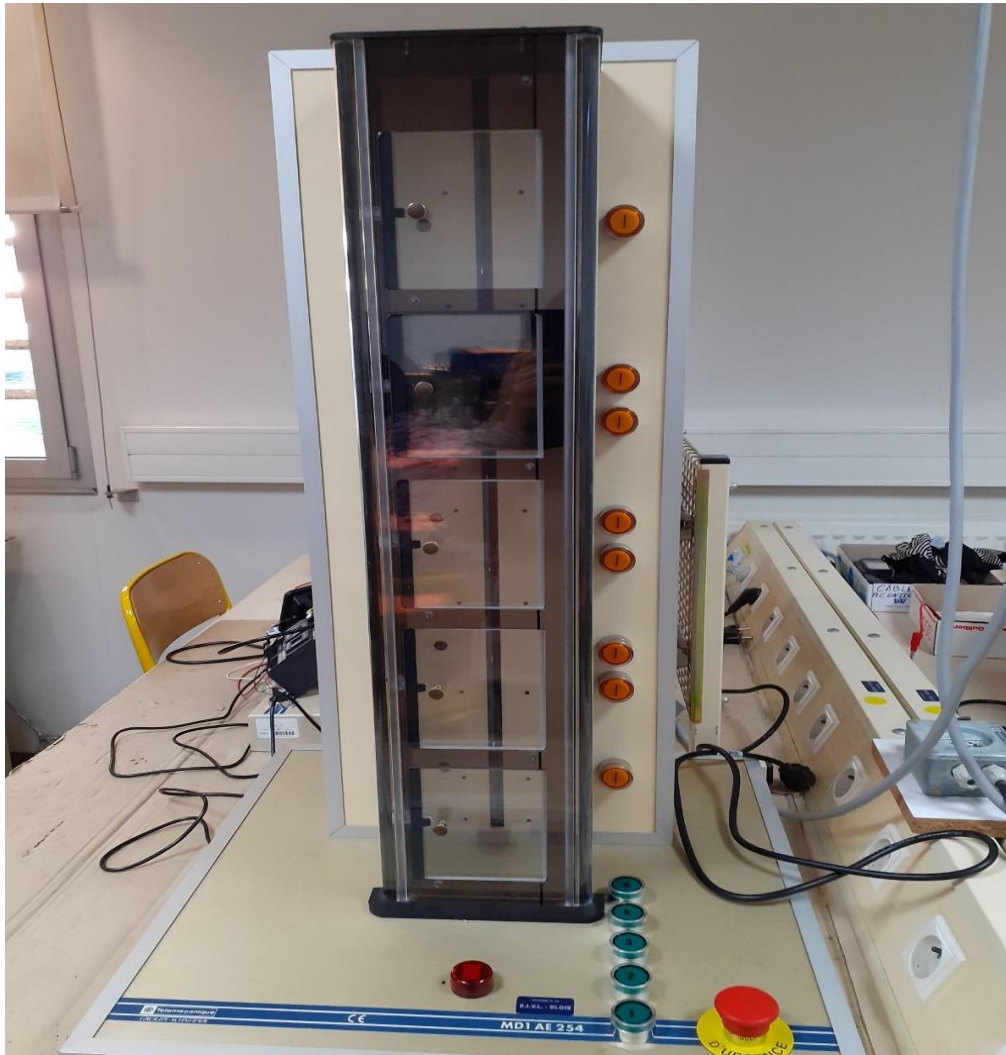
Le but de ce TP est de gérer ou de proposer un fonctionnement le mieux adapté selon nous pour un fonctionnement sans faille d'un ascenseur.

Dans ce projet, nous tiendrons compte uniquement à la situation qui nous est proposée. Nous parlerons tout de même d'un cas général à la fin de notre rapport notamment pour la gestion de priorités.

### Le diagramme de Gantt:

TÂCHES	Semaine 43	Semaine 46	Semaine 47	Semaine 48	Semaine 50
Prise de connaissance du projet et établissement de la GEMMA					
Identification des entrées et des sorties					
Echange d'idées et lancement du programme simple					
Réalisation des fonctions manuelle, automatique, principal					
Fonction de sécurité et l'affichage ( 7 segments)					
Les priorités					
Rapport					

## **L'ASCENSEUR:**



Nous avons un ascenseur à cinq (5) étages dont le premier se trouve au rez-de-chaussée.

### **APPEL EXTERIEUR :**

Un utilisateur, à l'extérieur, peut appeler la cabine grâce à des boutons d'appel (boutons en orange) : BPME1, BPME2, BPDE2, BPME3, BPDE3, BPME4, BPDE4, BPDE5 correspondant respectivement aux montées et descentes des étages 1,2,3,4 et 5.

### **APPEL INTERIEUR :**

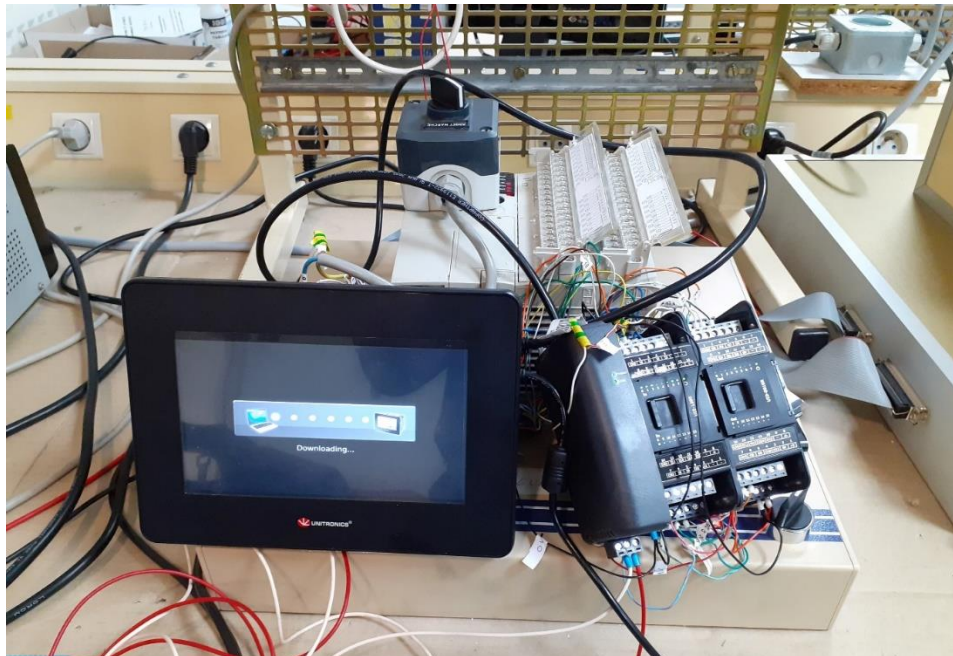
A l'intérieur de la cabine, on a cinq (5) boutons d'appel intérieur (boutons en vert) BPAC1, BPAC2, BPAC3, BPAC4, BPAC5.

## LES CAPTEURS :

Trois types de détection sont utilisés :

- Détection du positionnement aux étages.
- Détection de la fermeture de la porte à chaque étage.
- Capteurs de fin de course haut et bas de sécurité : FDCHAUT et FDCBAS sont câblés sur la partie opérative et sur l'Automatiquemate.

## LES OUTIL :

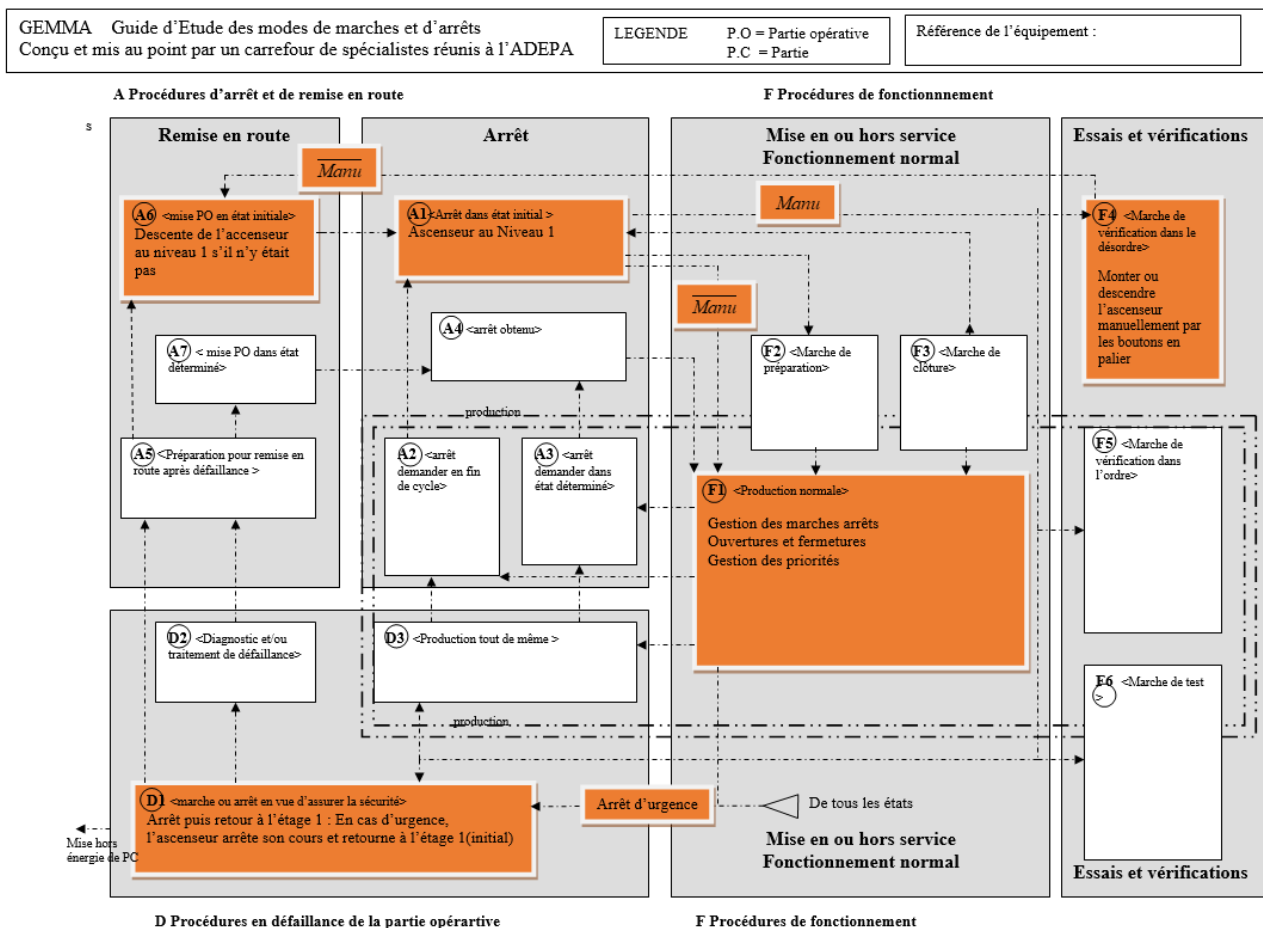


Nous utilisons le modèle de contrôleur (US / 7-B10-RA28) en association avec UAG-CX-XKPXXX qui a 2 portes input et output (UID-1600\_0 et UID 0016R\_1).

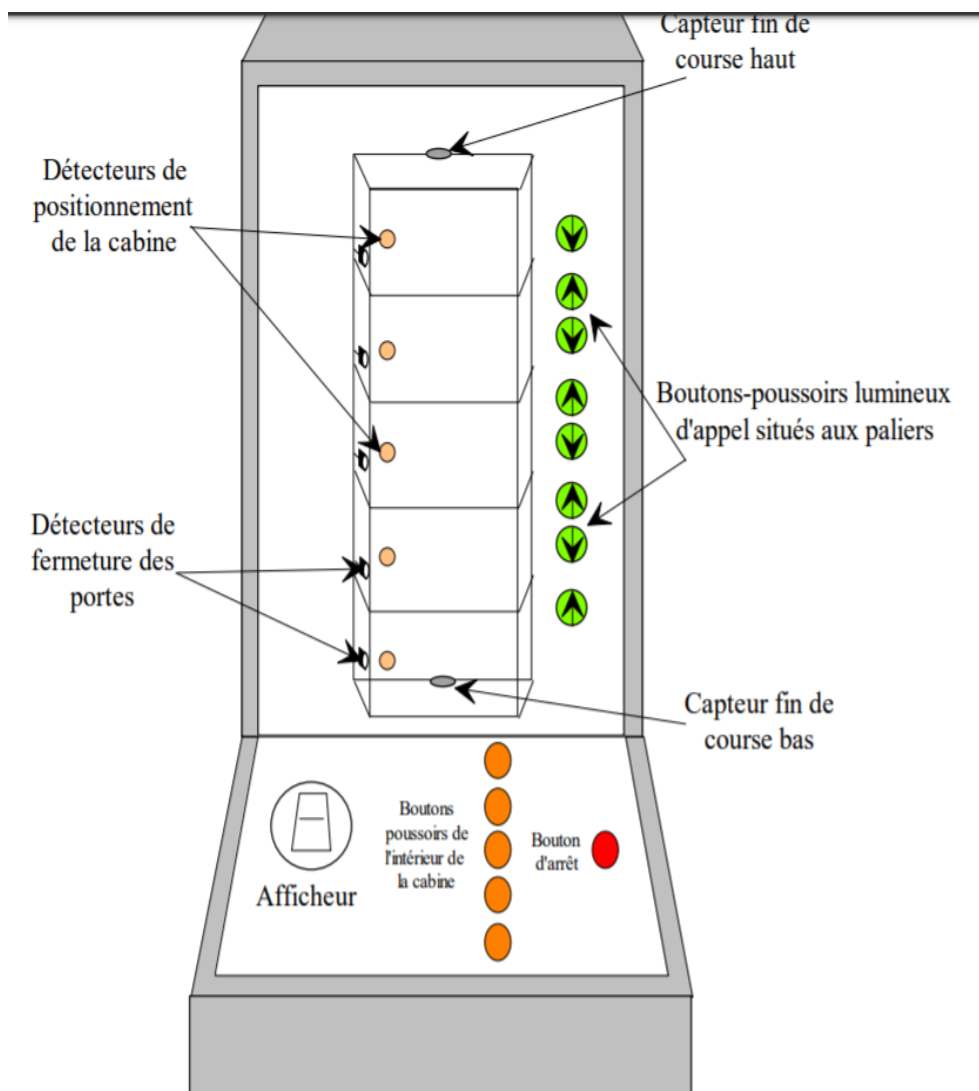
## II. Les schéma :

### 1. Schéma de GEMMA :

GEMMA comme son nom l'indique c'est le guide défini par les programmeurs(nous) et qui montre la manière dont ils voudraient que l'outil fonctionne (L'état initial, la maintenance ou non, les urgences,...)

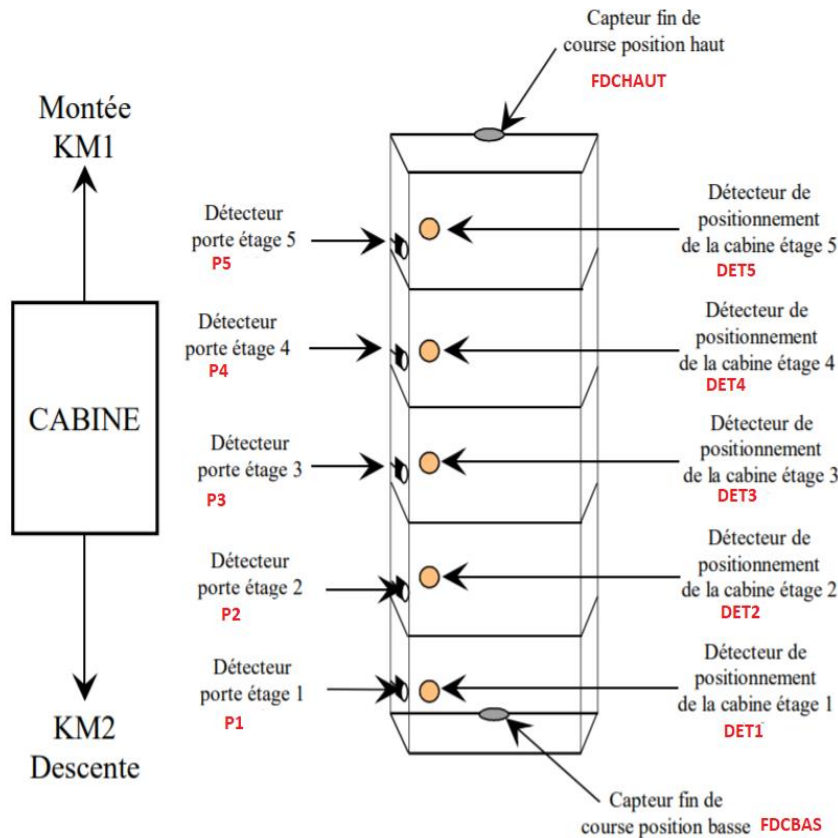


## 2. Schéma de la maquette :





### 3. Schéma des capteurs et des actionneurs :



#### Les capteurs :

- On a au total 12 capteurs répartis en 3 catégories (cabine, porte, course).
- Cabine : 5 pour la détection de la cabine.
- Portes : 5 pour la détection de la fermeture des portes.
- Course : 1 pour la détection de fin de course haut.  
1 pour la détection de fin de course bas.

#### Détection cabine :

La cabine sert à effectuer deux mouvements: monter et descendre. Elle s'arrête lorsqu'elle arrive au niveau demandé ou voulu. La présence de la cabine à un étage donné est détectée par un capteur DET1, DET2, DET3, DET4, DET5.

### Détection fermeture des portes :

A chaque arrêt de l'ascenseur à un étage donné, on attend l'ouverture des portes détectée par les capteurs P1, P2, P3, P4, P5. Au bout de quelques secondes si les portes sont refermées, alors nous on démarre vers l'appel suivant.

### Détection fin de course :

Pour la sécurité de l'utilisateur, il existe des capteurs FDCHAUT et FDCBAS respectivement fin de course haut et fin de course bas.

## III. Introduction du logiciel :

Pour rappel, nous nous sommes servis du logiciel libre UniLogic.

### Les entrées:

IO ▶ UID-1600\_0 ▶ Inputs

#	Name	Type	Format	Alias Name	Description
0	<a href="#">Inputs_0</a>	BIT	Binary	P1	Porte_etage1
1	<a href="#">Inputs_1</a>	BIT	Binary	P2	Porte_etage2
2	<a href="#">Inputs_2</a>	BIT	Binary	BPugt	Boutton_Urgent
3	<a href="#">Inputs_3</a>	BIT	Binary	BPAC1	Appel_cabine1
4	<a href="#">Inputs_4</a>	BIT	Binary	BPAC2	Appel_cabile2
5	<a href="#">Inputs_5</a>	BIT	Binary	BPAC3	Appel_cabile3
6	<a href="#">Inputs_6</a>	BIT	Binary	P3	Porte_etage3
7	<a href="#">Inputs_7</a>	BIT	Binary	P4	Porte_etage4
8	<a href="#">Inputs_8</a>	BIT	Binary	P5	Porte_etage5
9	<a href="#">Inputs_9</a>	BIT	Binary	BPME1	Monter_etage1
10	<a href="#">Inputs_10</a>	BIT	Binary	BPME2	Monter_etage2
11	<a href="#">Inputs_11</a>	BIT	Binary	BPME3	Monter_etage3
12	<a href="#">Inputs_12</a>	BIT	Binary	BPME4	Monter_etage4
13	<a href="#">Inputs_13</a>	BIT	Binary		
14	<a href="#">Inputs_14</a>	BIT	Binary		
15	<a href="#">Inputs_15</a>	BIT	Binary		

## IO ▶ On board IO ▶ Digital Inputs

#	Name	Type	Format	Alias Name	Description
0	<a href="#">Digital Inputs_0</a>	BIT	Binary	BPDE2	Descendre etage 2
1	<a href="#">Digital Inputs_1</a>	BIT	Binary	FDCHAUT	FDC EN HAUT
2	<a href="#">Digital Inputs_2</a>	BIT	Binary	DET1	Detecteur etape 1
3	<a href="#">Digital Inputs_3</a>	BIT	Binary	DET2	Detecteur etape 2
4	<a href="#">Digital Inputs_4</a>	BIT	Binary	DET3	Detecteur etape 3
5	<a href="#">Digital Inputs_5</a>	BIT	Binary	DET4	Detecteur etape 4
6	<a href="#">Digital Inputs_6</a>	BIT	Binary	DET5	Detecteur etape 5
7	<a href="#">Digital Inputs_7</a>	BIT	Binary	FDCBAS	FDC EN BAS
8	<a href="#">Digital Inputs_8</a>	BIT	Binary	MNAUTO	MANUEL AUTO
9	<a href="#">Digital Inputs_9</a>	BIT	Binary	BPDE3	Descendre_etage 3
10	<a href="#">Digital Inputs_10</a>	BIT	Binary	BPDE4	Descendre_etage 4
11	<a href="#">Digital Inputs_11</a>	BIT	Binary	BPDE5	Descendre_etage 5
12	<a href="#">Digital Inputs_12</a>	BIT	Binary	BPAC4	Appel_cabine4
13	<a href="#">Digital Inputs_13</a>	BIT	Binary	BPAC5	Appel_cabine5

## Les sorties :

## IO ▶ UID-0016R\_1 ▶ Outputs

#	Name	Type	Format	Alias Name	Description
0	<a href="#">Outputs_0</a>	BIT	Binary	VPH3	voyant dappel palier etage 3 haut
1	<a href="#">Outputs_1</a>	BIT	Binary	VPH4	voyant dappel palier etage 4 haut
2	<a href="#">Outputs_2</a>	BIT	Binary	VC5	voyant dappel cabine etage 5
3	<a href="#">Outputs_3</a>	BIT	Binary	VPB5	voyant dappel palier etage 5 bas
4	<a href="#">Outputs_4</a>	BIT	Binary	S1	afficheur droit poids faible 1
5	<a href="#">Outputs_5</a>	BIT	Binary	S2	afficheur droit 2
6	<a href="#">Outputs_6</a>	BIT	Binary	S4	afficheur droit 4
7	<a href="#">Outputs_7</a>	BIT	Binary	S8	afficheur droit poids fort 8
8	<a href="#">Outputs_8</a>	BIT	Binary		
9	<a href="#">Outputs_9</a>	BIT	Binary		
10	<a href="#">Outputs_10</a>	BIT	Binary		
11	<a href="#">Outputs_11</a>	BIT	Binary		
12	<a href="#">Outputs_12</a>	BIT	Binary	VPB2	voyant dappel palier etage 2 bas
13	<a href="#">Outputs_13</a>	BIT	Binary	VPB3	voyant dappel palier etage 3 bas
14	<a href="#">Outputs_14</a>	BIT	Binary	VPB4	voyant dappel palier etage 4 bas
15	<a href="#">Outputs_15</a>	BIT	Binary		

IO ▶ On board IO ▶ Digital Outputs

#	Name	Type	Format	Alias Name	Description
0	<a href="#">Digital Outputs_0</a>	BIT	Binary	MC	monter_cabine
1	<a href="#">Digital Outputs_1</a>	BIT	Binary	DC	descendre_cabine
2	<a href="#">Digital Outputs_2</a>	BIT	Binary	VC1	voyant dappel cabine etage 1
3	<a href="#">Digital Outputs_3</a>	BIT	Binary	VC2	voyant dappel cabine etage 2
4	<a href="#">Digital Outputs_4</a>	BIT	Binary	VC3	voyant dappel cabine etage 3
5	<a href="#">Digital Outputs_5</a>	BIT	Binary	VC4	voyant dappel cabine etage 4
6	<a href="#">Digital Outputs_6</a>	BIT	Binary	VPH1	voyant dappel palier etage 1 haut
7	<a href="#">Digital Outputs_7</a>	BIT	Binary	VPH2	voyant dappel palier etage 2 haut

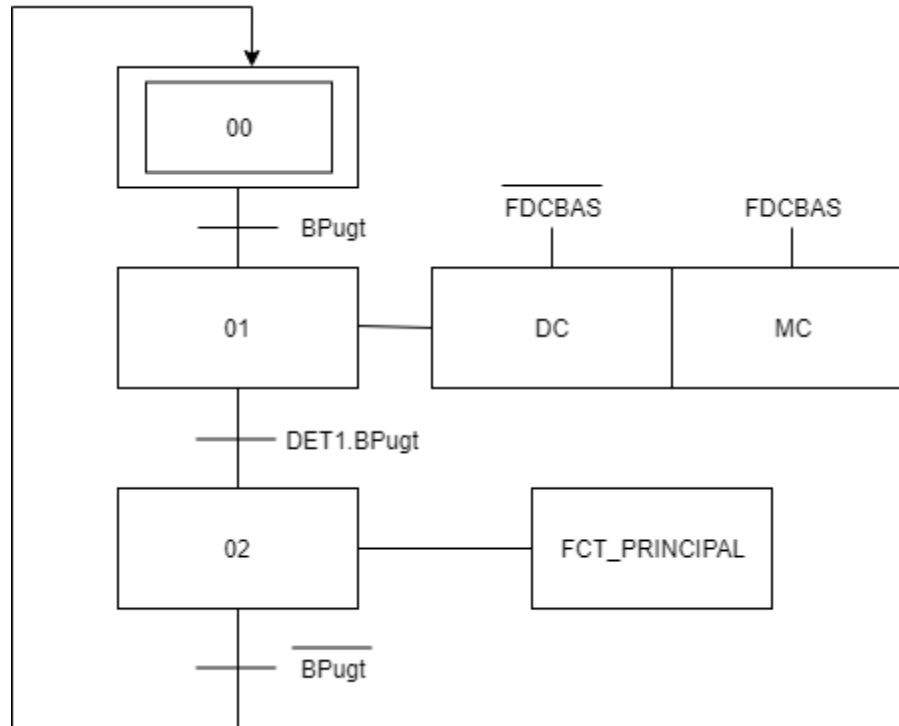
## Les variables globales:

Name	Type	Power-up/Initial	Format	Retain	Description
<a href="#">X30</a>	BIT	1	Binary	<input type="checkbox"/>	variable pour identifier l-étage du programme . Ici c'est éta
<a href="#">X31</a>	BIT		Binary	<input type="checkbox"/>	étage 31 du programme
<a href="#">X32</a>	BIT		Binary	<input type="checkbox"/>	étage 32 du programme
<a href="#">X20</a>	BIT	1	Binary	<input type="checkbox"/>	étage 20 du programme
<a href="#">X21</a>	BIT		Binary	<input type="checkbox"/>	étage 21 du programme
<a href="#">X22</a>	BIT		Binary	<input type="checkbox"/>	étage 22 du programme
<a href="#">X33</a>	BIT		Binary	<input type="checkbox"/>	étage 33 du programme
<a href="#">X34</a>	BIT		Binary	<input type="checkbox"/>	étage 34 du programme
<a href="#">X10</a>	BIT	1	Binary	<input type="checkbox"/>	étage 10 du programme
<a href="#">X11</a>	BIT		Binary	<input type="checkbox"/>	étage 11 du programme
<a href="#">X12</a>	BIT		Binary	<input type="checkbox"/>	étage 12 du programme
<a href="#">X00</a>	BIT	1	Binary	<input type="checkbox"/>	étage 00 du programme
<a href="#">X01</a>	BIT		Binary	<input type="checkbox"/>	étage 01 du programme
<a href="#">X02</a>	BIT		Binary	<input type="checkbox"/>	étage 02 du programme
<a href="#">X13</a>	BIT		Binary	<input type="checkbox"/>	étage 13 du programme
<a href="#">secu</a>	BIT		Binary	<input type="checkbox"/>	état des portes
<a href="#">etDemand</a>	INT8	1	Dec	<input type="checkbox"/>	la variable mémorise la 1er demande
<a href="#">etCurrent</a>	INT8	1	Dec	<input type="checkbox"/>	la variable mémorise l étage current
<a href="#">etAppel</a>	INT8	0	Dec	<input type="checkbox"/>	la variable mémorise la 2er demande
<a href="#">X31a</a>	BIT		Binary	<input type="checkbox"/>	étage 31a du programme
<a href="#">X31b</a>	BIT		Binary	<input type="checkbox"/>	étage 31b du programme
<a href="#">X32a</a>	BIT		Binary	<input type="checkbox"/>	étage 32a du programme
<a href="#">X32b</a>	BIT		Binary	<input type="checkbox"/>	étage 32b du programme
<a href="#">Bien Etage</a>	BIT	0	Binary	<input type="checkbox"/>	vérifie le cabin est en un étage ou pas

## IV. Programmation du grafcet:

Chaque étage du programme est correspondant à une variable globale qui commence par le lettre 'X' (Voir tableau de variable globales)

### 1. Grafcet de sécurité :



Ce grafcet présente une fonction de sécurité de l'ascenseur. S'il y a une demande d'arrêt urgent, l'ascenseur va être gelé. Après de n'avoir pas commandé le bouton arrêt urgent, le cabine réinitialisera au premier étage de l'ascenseur.

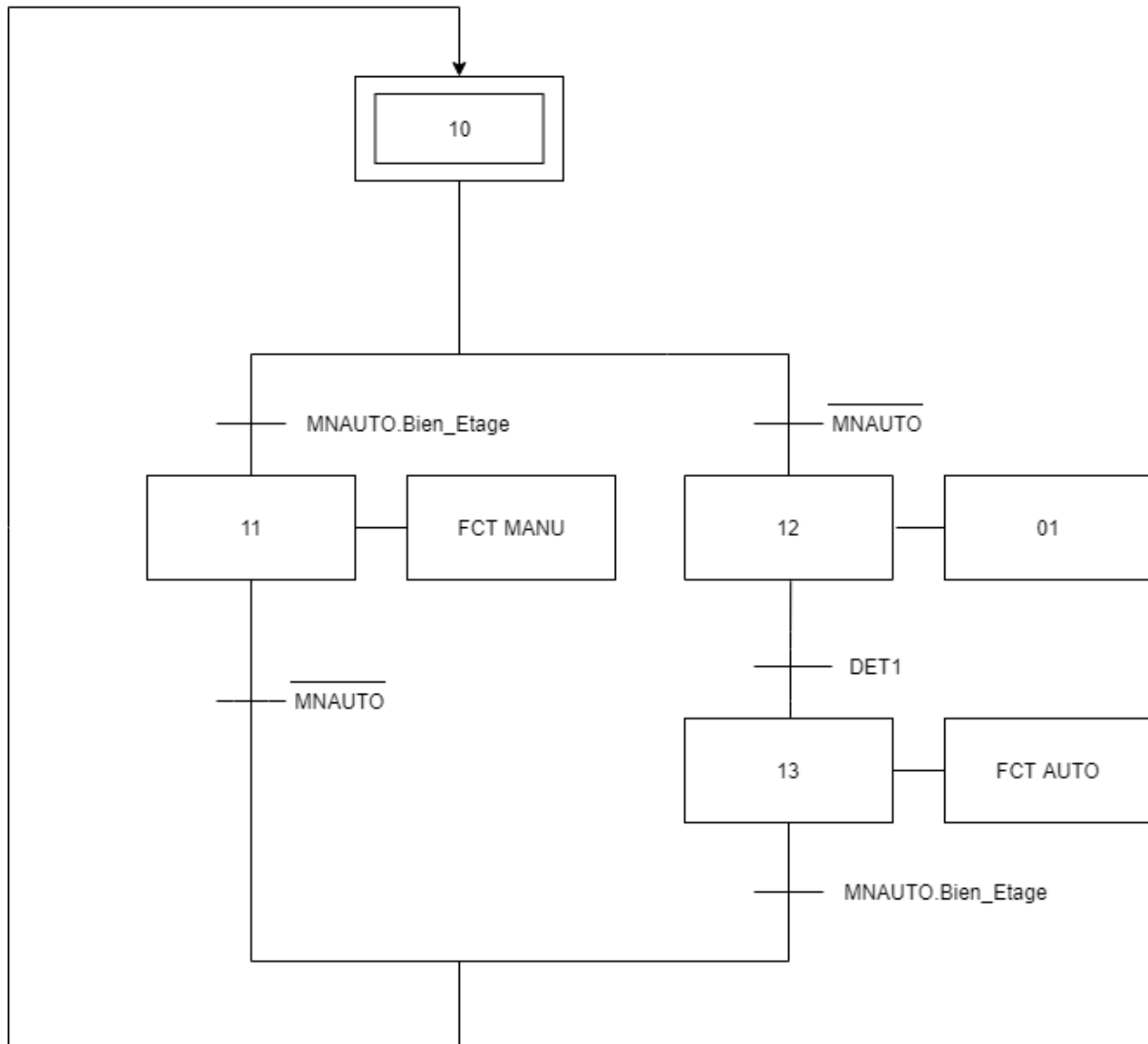
C'est la fonction principale que le système va exécuter premièrement quand nous installions l'ascenseur, et d'ici le système reste à cet étage si le bouton arrêt-urgent est commandé. Il contient 3 étage : X00, X01 et X02.

L'étage X00 est l'étage initial où le système va se situer s'il est gelé.

À l'étage X01, nous allons réinitialiser la cabine au premier étage de l'ascenseur. Et aussi dans cet étage nous allons réinitialiser les variables globales utilisées : Reset (VC1, VC2, VC3, VC4, VC5, S1, S2, S4, S8, VPB2, VPB3, VPB4, VPB5, VPH1, VPH2, VPH3, VPH4, X11, X12, X13, X21, X22, X31, X32) et Set ( X10, X20, X30).

Si la cabine est en premier et il n'y a pas le commande arrêt urgent, le système va passer à l'étage X02 où il appellera la fonction pour gérer les modes de fonction.

## 2. Grafcet principal :



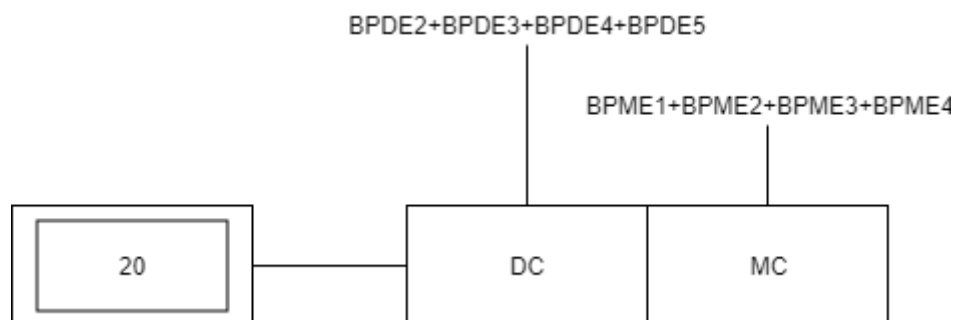
C'est un grafcet présentant la fonction de la gestion des modes de l'ascenseur en fonction de la commande du bouton "Manu/Auto". Il existe deux modes de fonction : mode <Maintenance> et mode <Automatique>. Au mode <Maintenance>, le système permet aux clients de monter ou descendre la cabine par les boutons d'appel au palier. En revanche, le mode <Automatique> permet de déplacer la cabine automatiquement d'après maximal deux demandes en même temps. Le fonctionnement de cette fonction est quand le système transfère du mode <Automatique> au <Maintenance>, la cabine ne va pas arrêter immédiatement, elle doit aller à l'étage le plus propre de l'ascenseur. Transférant du mode <Maintenance> au mode <Automatique>, le système réinitialisera à l'état initial (la cabine en premier étage de l'ascenseur).

En effet, ce grafcet contient 4 étage : X10, X11, X12 et X13. L'étage X10 est l'étage initial où le système va se situer quand cette fonction a été appelé. L'étage X11 est correspondant au mode <Maintenance>. L'étage X13 est d'appeler le mode <Automatique>. L'étage X12 appellera l'étage X01 de la fonction de sécurité afin de réinitialiser l'état de la cabine avant d'aller l'étage X13.

Particulièrement, pour être sûr que la position de la cabine est adaptée à transférer le mode, nous avons utilisé la variable <Bien\_etage>. Cette variable obtient le valeur '1' seulement si la cabine est au un étage déterminé. Imaginons que quand le système est au étage X13 et la cabine est en train de monter du 2ème étage au 3ème étage, nous change la commande du le bouton du mode, la cabine doit aller au 3ème étage pour que la valeur de <Bien\_etage> soit '1'. Au ce moment, le système transfère de l'étage X13 à l'étage X10 et accède l'étage X11. En bref, la variable Bien\_Etage aidera nous à détermier si le cabin est exactement au étage pour arrêter à l'étage le plus proche dans le sens du mouvement si l'ascenseur fonctionne.

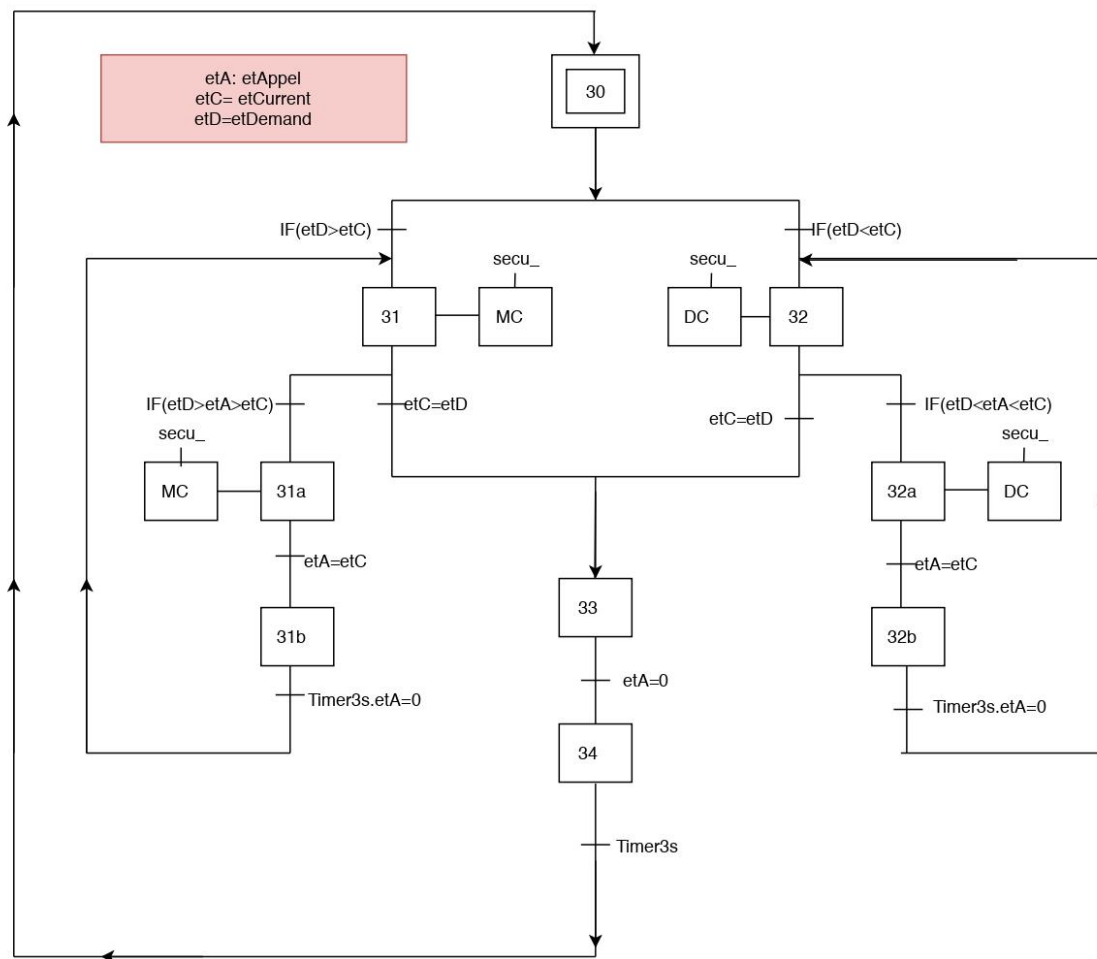
$\text{Bien\_etage} = (\text{DET1} + \text{DET2} + \text{DET3} + \text{DET4} + \text{DET5})$

### 3. Grafcet Manuel :



Ce grafcet présente la fonction <Maintenance> de l'ascenseur. Il est très simple et il contient seulement un étage X20, ce qui permet de monter ou descendre la cabine par les boutons d'appel au palier.

## 4. Grafcet Automatique :



Ce grafcet présente la fonction automatique de l'ascenseur avec la priorité. Dans ce projet, nous avons géré deux appels distancés dans le temps puis deux appels simultanés. Nous comprenons bien que dès qu'on a plusieurs appels en même temps, il va falloir faire un choix donc **définir des priorités**. Puisque nous n'avons que cinq étages et deux appels simultanés, nous avons donc décidé de donner la priorité aux demandes les plus proches de l'étage où se trouve la cabine. C'est-à-dire si la cabine monte du premier étage au 4<sup>ème</sup> étage et il y a une demande au 3<sup>ème</sup> étage, la cabine va l'arrêter en trois secondes. Si la deuxième demande est hors des étages que la cabine va passer. Nous allons le sauvegarder et la manipuler après.

Algorithme :

Nous avons utilisé 3 variables globales :

**etDemand** qui permet de gérer la demande qui est en train d'exécuter

**etAppel** qui permet de garder la deuxième demande en mémoire

**etCurrent** qui définit la position de la cabine



Pour permettre de manipuler deux demande en même temps, nous proposons une priorité aux demandes. La première demande que le système reçoit des boutons va être enregistré dans la variable 'etDemand' et être manipulé en premier. Mais si le système reçoit la deuxième demande au moment que la première demande exécutée, nous allons vérifier que l'étage de la deuxième est à l'intérieur ou à l'extérieur des étages que la cabine traversera pour manipuler la première demande. S'il est un des étages traversés, le système va faire la deuxième demande et après continuer de manipuler la première. S'il n'est pas un des étages traversés, le numéro de l'étage demandé va être gardé et le système attribuera la valeur de 'etAppel' à la variable 'etDemand' après d'avoir fini de faire la première demande. Et maintenant, la variable 'etAppel' est égal à 0, elle peut recevoir une autre demande pour comparer avec la demande qui est en cours de mise en œuvre. Et le système va continuer d'effectuer de tels cycle.

Expliquer le fonctionnement de grafcet Automatique, ce grafcet contient 10 étage : X30, X31, X31a, X31b, X32, X32a, X32b, X33 et X34.

L'étage X30 est l'étage initial où le système va se situer quand cette fonction a été appelé et il signifie aussi l'état que aucune demande (l'étage actuel est égal à l'étage demande).

L'étage X31 est de monter la cabine si l'étage demande est supérieur de l'étage actuel. L'étage X31a et X31b sont les sous-étages de l'étage X31 pour manipuler la deuxième demande s'il est un des étages traversés par la cabine. X31a continue à monter la cabine pour aller à l'étage sauvegardé dans la variable 'etAppel'. X31b permet d'attendre 3 secondes quand la cabine arrive.

L'étage X32, X32a et X32b ont la même opération mais pour les commandes de descente.

A l'étage X33, le système assigne la valeur de la variable 'etAppel' à la variable 'etDemande' et réinitialise la variable 'etAppel' à la valeur 0 pour permettre au système de recevoir la prochaine demande.

L'étage X34 du grafcet est pour que la cabine attend 3 secondes après d'avoir allé à l'étage qui est enregistré dans la variable 'etDemand'.

### **Le variable secu :**

L'ascenseur va arrêter tout de suite quand quelqu'un ouvre la porte. C'est la raison pour laquelle nous le mettre comme la condition pour monter ou descendre la cabine.

$\overline{P1P2P3P4P5} = \text{Reset}(\text{secu\_})$

## Afficheur 7 segments :

Il affiche le numéro de l'étage dans lequel se trouve la cabine :

Etage1: DET1 . Rset(S8) . Rset(S4) . Rset(S2) . Set(S1)	(0001=1)
Etage2: DET2 . Rset(S8) . Rset(S4) . Set(S2) . Rset(S1)	(0010=2)
Etage3: DET3 . Rset(S8) . Rset(S4) . Set(S2) . Set(S1)	(0011=3)
Etage4: DET4 . Rset(S8) . Set(S4) . Rset(S2) . Rset(S1)	(0100=4)
Etage5: DET5 . Rset(S8) . Set(S4) . Rset(S2) . Set(S1)	(0101=5)

## Lumière des boutons :

Donne la luminosité a un bouton lorsqu'il est appuyé. Les acronymes que vous trouverez dans cette partie sont les suivants :VC, BPAC, BPME, BPDE.

**VCx** = voyant d'appel cabine x

**VPHx** = voyant d'appel palier etage x haut

**VPBx** = voyant d'appel palier etage x bas

**BPAC** = bouton poussoir appel cabine

**BPME** = bouton poussoir monté étage

**BPDE** = bouton poussoir descente étage

$$VC1 = \overline{DET1}(VC1 + BPAC1)$$

$$VC2 = \overline{DET2}(VC2 + BPAC2)$$

$$VC3 = \overline{DET3}(VC3 + BPAC3)$$

$$VC4 = \overline{DET4}(VC4 + BPAC4)$$

$$VC5 = \overline{DET5}(VC5 + BPAC5)$$

$$VPH1 = \overline{DET1}(VPH1 + BPME1)$$

$$VPH2 = \overline{DET2}(VPH2 + BPME2)$$

$$VPH3 = \overline{DET3}(VPH3 + BPME3)$$

$$VPH4 = \overline{DET4}(VPH4 + BPME4)$$

$$VPB2 = \overline{DET2}(VPB2 + BPDE2)$$

$$VPB3 = \overline{DET3}(VPB3 + BPDE3)$$

$$VPB4 = \overline{DET4}(VPB4 + BPDE4)$$

$$VPB5 = \overline{DET5}(VPB5 + BPDE5)$$

## **CONCLUSION :**

Au sortir de ce projet, Nous avons appris beaucoup de chose : en premier la gestion d'un ascenseur est une chose réfléchi puisque mis à part le confort de l'utilisateur, sa sécurité est très importante. Aussi est-il important d'analyser toutes les situations et contextes avant de se lancer et trouver un bon algorithme permettre de résoudre tous les cas. En deuxième position, c'est un travail pas à pas. En faisant le schéma Gamma, nous avons pu obtenir une vue générale du fonctionnement de l'ascenseur. Cela nous a permis de construire facilement le graphe à faire et définir la relation entre ceux. Avec un diagramme de Gantt, nous avons trouvé une stratégie de travailler pas à pas pour pouvoir réussir ce projet. Finalement, grâce à ce projet, nous améliorons la capacité de travail en groupe.