REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



ECOLE NATIONALE POLYTECHNIQUE

Département d'Automatique
Spécialité Automatique

Projet de fin d'études

Pour l'obtention du diplôme
D'Ingénieur d'Etat en Automatique

THÈME

Conception d'un système de supervision pour un bâtiment intelligent :

Gestion de la distribution électrique, de l'éclairage et de la climatisation

Etudié par :

DADDI HAMMOU Abdennour
RAHMANI Mustapha Amine

Proposé et dirigé par :

M^{r}. A. BERKOUK
M^{r}. M.A. MESSAIF

Juin 2010

Ecole Nationale Polytechnique, 10, AV. Hassen Badi, El-Harrach, Algérie

REMERCIMENT

Nous tenons vivement à remercier notre promoteur le Pr.BERKOUK de l'Ecole Nationale Polytechnique pour son aide précieuse tout le long de notre projet de fin d'études.

Nous remercions également notre co-promoteur Mr M.A. MESSAIF de Schneider Electric qui nous a fait partager son expertise dans le domaine des Bâtiments Intelligents.

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptent d'évaluer notre projet.

Nous remercions également toute l'équipe du « Service Automatique » de Schneider Electric, et Moussab Ali en particulier pour leur disponibilité et leur soutien pendant notre stage de fin d'études.

Nous souhaitons aussi remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, et en particulier, Nos professeurs d'Automatique qui nous ont encadrés auparavant et tous nos enseignants pour les connaissances qu'ils nous ont transmis, leur disponibilité et leurs efforts.

Que tous ceux qui ont contribué de prés ou de loin à la réalisation de ce modeste travail trouvent ici l'expression de notre sincère gratitude.

Dédicaces

Je dédie ce modeste travail :

À Celle à qui mon cœur depuis ma naissance n’a pu qu’éprouver

qu’amour et reconnaissance, à celle qui a donné un sens à mon existence

À ma chère Mère.

À mon père, pour son amour et son soutien moral depuis mon enfance.

Père merci.

À la mémoire de mes chers grands pères.

À mes deux grands-mères pour leur amour et leur soutien

À mon frére pour son soutient tout au long de mon PFE

À toutes ma famille.

À mes meilleurs amis

et à tous ceux qui m’ont aidé de prés ou de loin dans la réalisation de ce travail.

RAHMANI Mustapha Amine

DEDICACES

Je dédie ce modeste travail tout d'abord à ma chère mère qui m'a tant guidé et inspiré, que Dieu te garde pour nous mère.

A mon cher père pour son soutien inconditionnel.

Pour toute ma famille ainsi que mes amis pour l'affection, l'égard et l'attention desquels ils ont fait preuve.

…A tous ceux qui connaissent et aiment Abdennour.

Abdennour DADDI HAMMOU

TABLE DES MATIÈRES

Introduction Générale 1
Plan de Travail 2

CHAPITRE I : Généralités sur les Bâtiments Intelligents

1. Les bâtiments intelligents 3
   1.1 Définition des bâtiments intelligents 3
      1.1.1. Définition se basant sur les performances 3
      1.1.2. Définition se basant sur les services 3
   1.2. Evolution des bâtiments intelligents 3
   1.3. Avantages des bâtiments intelligents 5
      1.3.1. Pour les occupants 5
      1.3.2. Pour le propriétaire du bâtiment 5
      1.3.3. Pour les entreprises de maintenance 5
      1.3.4. Du point de vue opérationnel 5
   1.4. Gestion Technique Centralisée (GTC) et Gestion Technique des Bâtiments (GTB) .. 5
2. Les différents équipements utilisés dans la GTB 6
   2.1. Les capteurs 6
   2.2. Les contrôleurs 9
3. Principales boucles de régulation et fonctions dans les bâtiments intelligents 14
   3.1. Contrôle de l'éclairage 14
   3.2. Contrôle de l'air ambiant 14
   3.3. Circuits thermiques 15
   3.4. Sécurité et Alarmes 15
4. Généralités sur les systèmes de Ventilation 16
   4.1 Les différents types de ventilation 16
   4.2. Notions foncdamentales sur la ventillation 18
   4.3 types de ventilateurs utilisés 21
5. Généralités sur les systèmes de climatisation 23

5.1 Système tout air, à débit constant, mono-gaine 23
5.2 Système tout air, à débit variable (VAV) 24
6. Efficacité énergétique 28
6.1. Avantages de l'efficacité énergétique 29
6.2. Les piliers de l'efficacité énergétique 29
6.3. Rôle de la GTB dans la promotion de l'efficacité énergétique 30
6.4. Empreinte de Schneider Electric dans le domaine de l'efficacité énergétique 30
6.4.1. Présentation de l'application HED 31
7. Conclusion 35

CHAPITRE II : Présentation du Logiciel Unity Pro

Présentation du logiciel Unity Pro 36
1. Introduction 38
2. Présentation de l'interface utilisateur d'Unity Pro 37
3. Création d'un nouveau projet sous Unity Pro 40
3.1. Configuration matérielle 40
3.2. Configuration d'un réseau 44
3.3. Définition et édition des variables 46
3.4. Structuration du programme sous Unity Pro 53
3.5. Edition du programme utilisateur 55
3.6. Compilation, simulation et transfert du programme vers l'automate 63
4. Exemple d'application 67
4.1. Cahier de charge 67
4.2. Traduction du GRAFCET en équations logiques 69
4.3. Programmation de l'exemple en FBD 69
4.3.1. La temporisation 69
4.3.2. Le régulateur PI 70
4. CONCLUSION 72

CHAITRE III Présentation du Logiciel SCADA Vijeo Citect

1. Introduction 73
2. Caractéristiques et avantages du logiciel Vijeo Citect 73

3. Outils et composantes de Vijeo Citect 73
3.1. Outils de configuration 74
3.2. Outils d'exécution 74
3.3. Les pilotes 75
4. Eléments d'un projet 75
4.1. Éléments graphiques 76
4.2. Les tags (Variables) 77
4.3. Alarmes 77
4.4. Éléments système 77
4.5. Éléments de communication 78
4.6. Eléments de serveurs d'E/S 78
5. Création et configuration d'un nouveau projet sous Vijeo Citect 79
5.1. Création d'un nouveau projet 79
5.2. Configuration du nouveau projet 79
5.3. Configuration de la communication 82
6. Les graphiques 83
6.1. Création de la page principale 84
6.2. Création des génies et des super génies: 86
7. Les évènements 95
8. Les alarmes 96
8.1 Exemple d'application 96
9. Les rapports 102
Conclusion 107

CHAPITRE IV: Application

1. Introduction et descriptif général de l'application 109
1.1. Description de la tour 109
2. Description de l'installation électrique 110
2.1 Le Poste de Livraison..... 111
2.2. Les transformateurs 113
2.3. Le TGBT 1 114
2.4. Le TGBT 2 116
2.5. Le Tableau Général du Courant Ondulé 116

2.6. Le Tableau divisionnaire d'éclairage 117
2.7. Le Tableau Général du Groupe Electrogène TGGE 118
2.8. Les alarmes 119
3. description du système d'éclairage 121
3.1. Description du système d'éclairage intérieur 122
3.2. Descriptif du système d'éclairage extérieur 126
4. Centrales de Traitement d'Air (CTA) 128
4.1 Présentation 128
4.2 Régulation de la température d'air repris 131
4.3 Création d'une DFB pour la régulation de la température d'air repris de la CTA .. 132
4.4 Gestion des alarmes de la CTA 134
4.5 Séquence de Marche/Arrêt de la CTA 135
5. Inventaire des capteurs et actionneurs utilisés pour chaque niveau 135
5.1 Sous Sol 135
5.2 Rez-de-chaussée 137
5.3 Premier Niveau 138
5.4 Deuxième niveau 140
5.5 Troisième niveau 140
5.6 Quatrième niveau 141
5.7 Cinquième niveau 141
5.8 Sixième niveau 141
5.9 Zones extérieures du bâtiment 141
6. Réseau de Communication du bâtiment 142
7. Conclusion 143

Conclusion Générale……...144

Perspectives ……145

Bibliographie……...146

LISTE DES FIGURES

Figure. I.1 : Evolution des bâtiments intelligents……...4

Figure. I.2 : Transmetteur de température..…...…….7

Figure. I.3 : Transmetteur de pression différentielle……..8

Figure. I.4 : sonde hygrostatique d'ambiance…….8

Figure. I.5 : capteur de lumière……...9

Figure. I.6 : système de commande décentralisé……...10

Figure. I.7 : Actionneur TOR commandé par automate……...10

Figure. I.8 : Entrées reçues par l'automate…….....11

Figure. I.9 : Architecture des API…….....11

Figure. I.10 : Cycle de traitement d'un programme…….....13

Figure. I.11 : Contrôleur IHC de Schneider…….....13

Figure. I.12 : Principe d'une ventilation naturelle……....16

Figure. I.13 : Principe d'une ventilation simple flux……....17

Figure. I.14 : Principe d'une ventilation double flux……....18

Figure. I.15 : Diagramme de pression totale pour un ventilateur…….....18

Figure. I.16 : Courbe caractéristique du réseau de distribution d'air……...19

Figure. I.17 : Courbes caractéristiques d'un ventilateur centrifuge à aubes inclinées vers l'arrièr…….19

Figure. I.18 : Point de fonctionnement d'un ventilateur……...20

Figure. I.19 : Les deux principaux types de ventilateurs……..21

Figure. I.20 : Les deux principaux types de ventilateurs centrifuges……...22

Figure. I.21 : Courbe caractéristique d'un ventilateur à aubes recourbées vers l'avant (AV) et à aubes recourbées vers l'arrière (AR), pour un même point de fonctionnement……22

Figure. I.22 : Variation du débit d'air en fonction de la vitesse de rotation du ventilateur.....23

Figure. I.23 : Exemple d'une CTA appliquée à un local……...24

Figure. I.24 : Exemple d'une configuration d'un bâtiment avec une CTA……...24

Figure. I.25 : Régulation locale du débit d'air pulsé……..…...25

Figure. I.26 : Exemple d'une régulation conjointe du débit d'air pulsé et repris……..26

Figure. I.27 : Exemple d'une régulation de pression dans un conduit d'air (solution1) …..26

Figure. I.28 : Exemple d'une régulation de pression dans un conduit d'air (solution2) …..27

Figure. I.29 : Exemple d'un système VAV mono-gaine……28

Figure. I.30 : Les trois piliers de l'efficacité énergétique……..30

Figure. I.31 : Vitrine technologique internationale de Schneider Electric…….31

Figure. I.32 : Consommation globale depuis 2009…….....32

Figure. I.33 : rapport quotidien de la consommation de l'énergie par heure…….32

Figure. I.34 : Pourcentage d'utilisation de l'énergie……..…...33

Figure. I.35 : Tendances mensuelles……..33

Figure. I.36 : Economies réalisés……...34

Figure. I.37 : Consommation d'énergie par étage et par service……...34

Figure. II.1 : Interface utilisateur……..37

Figure. II.2 : Interface utilisateur (suite) ….....38

Figure. II.3 : les différentes vues du navigateur de projet….....38

Figure. II.4 : vue structurelle du navigateur de projet….....39

Figure. II.5 : vue fonctionnelle du navigateur de projet….....39

Figure. II.6 : choix d'une CPU….....40

Figure. II.7 : remplacement d'une CPU….....41

Figure. II.8 : Ajout d'un rack….....42

Figure. II.9 : choix d'une alimentation….....42

Figure. II.10 : exemple de choix d'un module analogique….....43

Figure. II.11 : fenêtre décrivant le module d'entrée/sortie choisi….....43
Figure. II.12 : exemple de configuration d'une voie….....44
Figure. II.13 : création d'un nouveau réseau logique….....45
Figure. II.14 : configuration du réseau logique….....45
Figure. II.15 : association d'un réseau logique à un module de communication……....46
Figure. II.16 : familles de types de données….....47
Figure. II.17 : édition de variables élémentaires….....51
Figure. II.18 : édition de variables dérivées….....51
Figure. II.19 : édition des instances du bloc fonction élémentaire (EFB) ….....52
Figure. II.20 : édition des instances du bloc fonction dérivé (DFB) ….....52
Figure. II.21 : ajout du DFB à la Bibliothèque….....53
Figure. II.22 : ordre de priorité des différentes tâches….....53
Figure. II.23 : Création d'une nouvelle tâche….....54
Figure. II.24 : Création d'une nouvelle section….....55
Figure. II.25 : Exemple d'un programme écrit en FBD….....56
Figure. II.26 : Paramètres d'un bloc FBD….....57
Figure. II.27 : exemple d'un programme écrit en Ladder….....59
Figure. II.28 : éditeur de programme LD….....59
Figure. II.29 : exemple d'un programme écrit en SFC….....60
Figure. II.30 : éditeur de programme ST….....62
Figure. II.31 : génération du projet….....63
Figure. II.32 : génération sans erreurs….....63
Figure. II.33 : connexion de l'automate….....64
Figure. II.34 : transfert du projet vers l'automate….....64
Figure. II.35 : création d'une nouvelle table d'animation….....65
Figure. II.36 : Exemple d'une table d'animation….....65
Figure. II.37 : Représentation du bureau….....67

Figure. II.38 : GRAFCET de l'exemple du bureau….....67
Figure. II.39 : chronogramme de la temporisation AKF_TE….....69
Figure. II.40 : schéma fonctionnel du régulateur PI utilisé….....70

Figure. III.1 : Eléments d'un projet sous Vijeo Citect….....76
Figure. III.2 : Définition d'un nouveau cluster….....80
Figure. III.3 : Définition des serveurs sur Vijeo Citect….....81
Figure. III.4 : Choix du périphérique d'E/S. ….....82
Figure. III.5 : Configuration de l'adresse et du protocole de communication….....83
Figure. III.6 : Définition de tags….....85
Figure. III.7 : Création d'un nouveau génie….....87
Figure. III.8 : Ajout d'un texte sous le symbole du capteur….....88
Figure. III.9 : Edition de l'apparence du texte….....88
Figure. III.10 : popup permettant d'associer un tag à un génie….....89
Figure. III.11 : Programmation de la couleur du texte….....90
Figure. III.12 : Coller un génie dans la page principale….....91
Figure. III.13 : Configuration du génie….....92
Figure. III.14 : Configuration du bouton d'allumage dans le super génie….....93
Figure. III.15 : Configuration du bouton quitter dans le super génie….....94
Figure. III.16 : Lier un super génie à un génie….....95
Figure. III.17 : Aperçu de l'exécution de l'application, appel d'un super génie ….....95
Figure. III.18 : Configuration d'une rangée de luminaire….....96
Figure. III.19 : Configuration d'un évènement….....97
Figure. III.20 : Configuration de deux évènements afin de marquer l'intervalle de detection d'intrusion….....98
Figure. III.21 : Configuration d'une alarme digitale….....99
Figure. III.22 : Configuration d'une catégorie d'alarme….....100

Figure. III.23 : Exécution de l'application et affichage de l'alarme….....102
Figure. III.24 : Page d'alarme….....103
Figure. III.25 : Configuration d'un périphérique de sortie pour le rapport….....104
Figure. III.26 : Configuration des propriétés d'un rapport….....104
Figure. III.27 : Vue de la TGBT de l'étage 1 de l'exemple d'application….....106

Figure. IV.1 : Page d'Accueil du projet….....110
Figure. IV.2 : Synoptique général de l'installation électrique….....111
Figure. IV.3 : Schéma électrique du poste de livraison….....112
Figure. IV.4 : Exemple de la disposition des différentes cellules dans un poste de livraison….....113
Figure. IV.5 : Exemple de la disposition des transformateurs dans une tour….....113
Figure. IV.6 : Schéma électrique du tableau TGBT 1….....114
Figure. IV.7 : Exemple de disposition d'armoire dans un TGBT….....115
Figure. IV.8 : Schéma électrique du TGBT 2….....116
Figure. IV.9 : Tableau Général du Courant Ondulé….....117
Figure. IV.10 : Le tableau divisionnaire pour l'éclairage du premier étage….....118
Figure. IV.11 : Schéma électrique du Tableau Général du Groupe Electrogène….....118
Figure. IV.12 : Page d'alarme….....120
Figure. IV.13 : Page sommaire des alarmes….....120
Figure. IV.14 : Page de commande d'éclairage du premier étage….....122
Figure. IV.15 : Page de Gestion horaire du premier étage….....123
Figure. IV.16 : Vue de l'étage 1….....124
Figure. IV.17 : Grafcet du cahier de charge de l'éclairage de la zone 2B….....125
Figure. IV.18 : Gestion de l'éclairage de la zone 2B en fonction de la présence et de l'éclairage naturel ….....126
Figure. IV.19 : Tableau divisionnaire de l'éclairage extérieur….....127

Figure. IV.20 : Page de gestion de l'éclairage extérieur….....128

Figure. IV.21 : Accès à la page des CTA….....129

Figure. IV.22 : Choix d'une CTA ….....129

Figure. IV.23 : Synoptique d'une CTA….....130

Figure. IV.24 : Régulateur Split Range….....131

Figure. IV.25 : Schéma de régulation de la CTA….....132

Figure. IV.26 : DFB pour la régulation de la température d'air repris de la CTA…...133

Figure. IV.27 : Synoptique de la partie communication réseau….....143

LISTE DES Tableaux

Tableau. I.1 : Règles de similitude pour un ventilateur…….20

Tableau. II.1 : Familles de variables dans Unity Pro…… …47

Tableau. II.2 : variables internes …… ……...48

Tableau. II.3 : les constantes %K …… ……..49

Tableau. II.4 : les constantes %I …… ……49

Tableau. II.5 : variables système …… ……49

Tableau. II.6 : variables réseau ……...…50

Tableau. II.7 : variables d'Entrée/Sortie ……...…50

Tableau. II.8: Structure d'une instruction en LIST …… …61

Tableau. II.9 : Paramètres de la temporisation AKF_TE……69

Tableau. II.2 : Paramètres du régulateur PI utilisé …… …70

Tableau. III.1 : liste des tags utilisés……86

Tableau. IV.1 : inventaire de l'installation électrique située au sous sol ……..……136

Tableau. IV.2 : Capteur et actionneurs de la CTA 1 pour le Sous Sol et le rez-de-chaussée……..136

Tableau. IV.3 : Tableau. IV.3 Nombre et type de points d'E/S pour le Sous Sol…….136

Tableau. IV.4 : Choix des Embases de Momentum pour le Sous Sol ……..……..…137

Tableau. IV.5 : inventaire des équipements du Tableau Divisionnaire d'Eclairage du Rez-de-chaussée ……..…….137

Tableau. IV.6 : Nombre et type de points d'E/S pour le rez-de-chaussée ……..…..137

Tableau. IV.7 : Choix des Embases de Momentum pour le rez-de-chaussée ……138

Tableau. IV.8 : inventaire des équipements du Tableau Divisionnaire d'Eclairage du premier niveau ……..……138

Tableau. IV.9 : Capteur et actionneurs de la CTA 2 pour le premier et le second niveau…….138

Tableau. IV.10 : Capteur et actionneurs pour l'éclairage de la Zone 2B ……....…139

Tableau. IV.11 : Nombre et type de points d'E/S pour le Premier niveau ……139

Tableau. IV.12 : Choix des Embases de Momentum pour le Premier niveau ….……139

Tableau. IV.13 : Choix des Embases de Momentum pour le Second niveau ……..140

Tableau. IV.14 : Choix des Embases de Momentum pour le Troisième niveau ……..140

Tableau. IV.15 : Equipements pour les zones extérieures ……..……141

Tableau. IV.16 : Nombre et type de points d'E/S pour les zones extérieures ……..141

Tableau. IV.17 : N Choix des Embases de Momentum pour les zones extérieures ….142

Introduction Générale

Introduction Générale

Le bâtiment est l'un des secteurs qui consomme le plus d'énergie et ceci au niveau mondial. De ce fait, il constitue l'un des principaux émetteurs de gaz à effet de serre. De plus une consommation irrationnelle en énergie fait très vite grimper la facture énergétique qui provient essentiellement de l'éclairage, de la ventilation et de la climatisation. Dans le but d'améliorer l'efficacité énergétique des bâtiment, le concept de « Bâtiment Intelligent » a été crée.

Le bâtiment intelligent fait appel à diverses technologies pour améliorer l'environnement et la fonctionnalité du bâtiment, tout en contrôlant les coûts d'exploitation. Ainsi, un système efficace énergétiquement fournit l'énergie au plus faible coût et évite le gaspillage d'énergie grâce à la gestion des aires occupées.

Le domaine de la Gestion Technique des Bâtiments étant d'actualité, nous avons décidé de l'explorer et d'y apporter notre contribution en développant une application de supervision de l'installation électrique, d'éclairage (intérieur et extérieur) et de climatisation pour un bâtiment à six étages avec rez-de-chaussée et sous sol.

Pour se faire, nous avons effectué un stage chez l'un des leaders mondiaux dans le domaine des bâtiments intelligents qui est Schneider Electric.

Notre application s'articulant autour des automates programmables, nous avons débuté notre étude par un apprentissage du logiciel **Unity Pro** pour la programmation des automates programmables de la gamme Schneider Electric puis nous nous sommes familiarisé avec le logiciel **Vijeo Citect** qui est l'un des logiciels de conception de systèmes de supervision les plus connus et les plus utilisés par les professionnels de l'automatisation.

Plan de Travail

Nous avons organisé notre projet en quatre chapitres :

Le premier chapitre donne des généralités sur les bâtiments intelligents en explicitant les concepts de base de ce domaine qui nous aideront à mieux appréhender notre application.

Le deuxième chapitre est consacré au logiciel Unity Pro pour la programmation des automates programmables de la gamme Schneider Electric. Les différentes étapes de création d'un projet sous ce logiciel y sont clairement présentées et finalisées par un exemple d'application.

Le troisième chapitre traite quand à lui du logiciel de conception de systèmes de supervision Vijeo Citect. Là encore les différentes étapes de création d'un projet y sont définies et illustrées par un exemple d'application.

Dans le quatrième et dernier chapitre, nous avons exposé notre application de supervision de l'installation électrique, de l'éclairage et de la climatisation d'un bâtiment de six étages avec sous sol et rez-de-chaussée.

CHAPITRE I

Généralités sur les Bâtiments Intelligents

1. Les bâtiments intelligents

Le concept du bâtiment intelligent a reçu une grande attention de la part des chercheurs et des technologues durant les deux dernières décennies et plusieurs définitions du bâtiment intelligent (**B.I**) ont été suggérées, du fait du développement de l'industrie du bâtiment et de la technologie de l'information.

1.1 Définition des bâtiments intelligents

Il est difficile de donner une définition standard de ce qu'est un B.I, cependant il existe deux grandes approches débouchant chacune sur une définition :

1-Définition se basant sur les performances.
2-Définition se basant sur les services.

1.1.1. Définition se basant sur les performances

Cette approche définit les B.I en étalant les performances que ce dernier doit avoir.
Parmi ces définitions, on trouve celle du groupe **EIBG** (European Intelligent Building Group) situé au Royaume Uni qui dit qu'un B.I est crée pour donner à son utilisateur l'environnement le plus efficace possible en utilisant et en gérant de manière optimale les ressources et en minimisant les couts des équipements. [B1]

1.1.2. Définition se basant sur les services

Cette approche décrit les B.I du point de vue des services offerts et de leur qualité. Ainsi le **JIBI** (Japanese Intelligent Building Institute) donne la définition suivante :
Un B.I est un bâtiment qui fournit les fonctions de communication et d'automatisation des locaux, bureaux et bâtiments en assurant :
1-réception, traitement et transmission de l'information tout en assurant une gestion et une supervision efficace.
2- la satisfaction des personnes travaillant à l'intérieur du bâtiment.
3- réponses rapides, flexibles et économiques face aux changements internes et externes de l'environnement [B1].

1.2. Evolution des bâtiments intelligents

L'évolution des bâtiments intelligents est illustrée par la pyramide de la **figure I.1**

Cette pyramide est ouverte au sommet accentuant ainsi sur le fait que les informations (les données) et les systèmes intelligents de bâtiments peuvent être distants (c'est-à-dire situés dans d'autres bâtiments) et que la communication se fait via internet.

- Avant 1980, l'automatisation des bâtiments était achevée sur un seul appareil.

- Entre 1980 et 1985 tout les sous-systèmes tel que le contrôle d'accès, Chauffage-Ventilation-Air conditionné (HVAC pour Heating, Ventilation and Air- Conditioning), traitement électronique des donnée (EDP pour Electronic Data Processing), étaient intégré au niveau de sous fonctions individuelles (systèmes dédiés) et la communication entre les divers sous systèmes était impossible.
- Entre 1985 et 1990 l'intégration de sous systèmes de même nature ou de fonctions similaires était achevée.

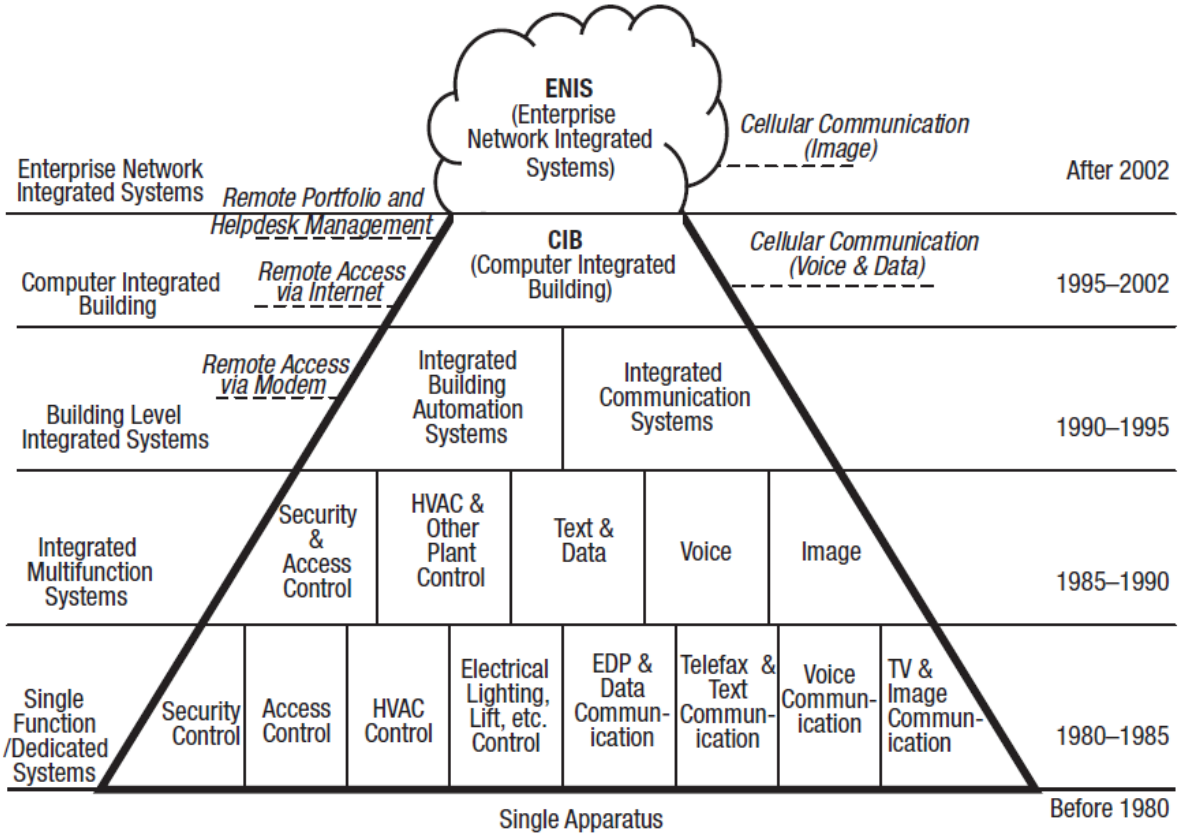


Fig. I.1 Evolution des bâtiments intelligents [B1]

- Entre 1990 et 1995 il y a eu intégration de tout les sous systèmes d'automatisation du bâtiment ainsi que tous les systèmes de communication au niveau du bâtiment. L'accès au système d'automatisation du bâtiment se faisait grâce au réseau téléphonique via un modem.
- Entre 1995 et 2002 la supervision du bâtiment pouvait se faire à distance et ceci grâce au développement du réseau Internet et du protocole IP (Internet Protocol).
- Après 2002 la supervision d'une entreprise toute entière (ayant des locaux distribués) et même d'une ville était possible grâce au développement exponentiel des technologies de l'information comme les services WEB, le XML… [B1]

1.3. Avantages des bâtiments intelligents [W2]

1.3.1. Pour les occupants

-Excellent contrôle des conditions internes de confort.
-Possibilité de contrôle individuel des pièces.
-Augmentation de la productivité des équipes de travail.
-Contrôle effectif de la consommation d'énergie.
-Améliorer la fiabilité du système (bâtiment) et la qualité de vie.

1.3.2. Pour le propriétaire du bâtiment

-Augmentation de la rentabilité du bâtiment.
-Facturation individuelle pour les locataires plus aisée.
-Contrôle local ou à distance du bâtiment.

1.3.3. Pour les entreprises de maintenance

-Disponibilité de l'information et diagnostic aisé.
-Etablissement d'un programme d'entretient informatisé
-Augmentation de l'efficacité des équipes de maintenance.
-Détection rapide des problèmes.
-Plus grande satisfaction des occupants.

1.3.4. Du point de vue opérationnel

-Diminution des couts des opérations.
-Utilisation efficace des ressources et services du bâtiment.
-Grande productivité.
-Diagnostic des défauts et indications rapide des alarmes.
-Excellente schématisation et documentation du système.

1.4. Gestion Technique Centralisée (GTC) et Gestion Technique des Bâtiments (GTB)

Nous allons maintenant définir ce que c'est la Gestion Technique des Bâtiments (GTB) et la Gestion Technique Centralisée (GTC) [W3]

1.4.1. Système de Gestion Technique Centralisée (GTC)

C'est le système de traitement des informations d'un seul domaine technique (chauffage, ou éclairages, ou climatisation, etc.) provenant d'un même site qui utilise généralement un réseau de communication propriétaire (propre au constructeur).

1.4.2. Système de Gestion Technique des Bâtiments (GTB)

C'est un système informatique généralement installé dans de grands bâtiments afin de superviser **l'ensemble** des équipements qui y sont installés par opposition à la GTC.

2. Les différents équipements utilisés dans la GTB

Dans cette section, nous allons exposer brièvement les différents systèmes électroniques et informatiques utilisés lors de la mise en œuvre d'un système de gestion des bâtiments.

La réalisation d'un système de gestion technique des bâtiments nécessite l'utilisation de **capteurs** afin de mesurer les différentes grandeurs physiques à commander, l'information mesurée est ensuite transmise vers un **calculateur** à travers un **bus de communication**, le calculateur est chargé d'exécuter un algorithme et de calculer l'ordre à transmettre vers **l'actionneur** qui doit agir sur le système à commander.

2.1. Les capteurs

2.1.1. Description générale

Le capteur constitue un élément fondamental dans une boucle de commande. Il est constitué des éléments suivant :

1. *Le corps d'épreuve :* c'est l'élément qui réagit à la mesurande (la grandeur à mesurer) et ceci en transformant les variations de la mesurande en variation d'une autre grandeur physique mesurable (déplacement, rotation, résistance,…etc.).
2. *Le corps de transduction :* il traduit la réaction du corps d'épreuve en un signal électrique mesurable (courant, tension, résistance…)
3. *Le module de conditionnement :* il a pour rôle le traitement du signal (filtrage, correction,…etc.), la mise en forme et l'amplification du signal électrique ainsi que l'alimentation du capteur.

2.1.2. Types de capteurs

Les capteurs peuvent être classifiés en :

1. *Capteur analogique* : le signal de sortie étant analogique continu, il prend des valeurs soit entre 0-10V, ou entre 0-20mA ou encore entre 4-20mA. Un capteur analogique peut être soit :

a. *Capteur passif :* consistant exclusivement en un corps d'épreuve et ne contenant pas de corps de transduction, le conditionnement du signal étant assuré par le calculateur, ainsi, aucune **puissance** d'alimentation n'est requise pour ce type de capteur, un exemple est le capteur de température avec sortie de type résistance.
b. *Capteur actif :* C'est un capteur qui prélève l'énergie nécessaire à son fonctionnement du signal à mesurer.

2. *Capteur numérique :* le signal de sortie est directement codé sous forme numérique, on distingue en particulier les capteurs 'tout ou rien' ou TOR qui donnent une sortie ayant deux états (on/off) selon que le signal mesuré est au dessus ou en dessous d'une valeur seuil, on peut citer comme exemple les thermostats, les fins de course, les pressostats,…etc.

2.1.3. Principaux capteurs utilisés en GTB

Dans le domaine de la GTB, on s'intéresse à la régulation de différentes grandeurs physiques, et par suite plusieurs types de capteurs sont utilisés. Parmi les principaux capteurs utilisés on peut citer :

1. *Les sondes de température de l'air:* utilisés notamment pour une régulation énergétiquement efficace des installations de chauffage, ventilation et climatisation, on peut citer comme exemple le transmetteur EGT 130, ayant un signal de sortie de 0-10V et constitué d'un élément résistif en Nickel, la variation de température se traduit en une augmentation de la résistance, une électronique est chargé ensuite de convertir cette variation de résistance en un signal normalisé 0-10V [F1]



Fig. I.2 Transmetteur de température [F1]

2. *Les transmetteurs de pression et débit d'air :* utilisé pour plusieurs applications, en particulier pour la mesure de la pression dans les conduits dans les installations de ventilation. On peut citer comme exemple le SDU : transmetteur de faible pression différentielle, permettant une mesure de pression différentielle jusqu'à 10 mbars et délivrant un signal de sorite de 0-10V [F2].

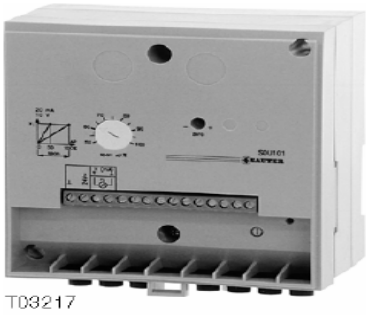


Fig. I.3 Transmetteur de pression différentielle [F2]

3. *Les sondes d'humidité :* où sondes hygrostatiques d'ambiance, elles sont utilisées dans les applications de régulation d'humidité de l'air et dans les applications de chauffage, ventilation et climatisation. On peut citer comme exemple le capteur EGH120 qui est un capteur d'humidité relative délivrant un signal en sortie de 0-10 V ou 4-20 mA. Ce transmetteur comporte aussi un ajusteur qui permet de compenser les dérivées permanentes pouvant apparaitre lorsque le capteur travaille dans un environnement hostile comportant des gaz agressifs [F3]



Fig. I.4 sonde hygrostatique d'ambiance [F3]

4. *Les capteurs de lumière :* utilisés pour un asservissement de l'éclairage intérieur en prenant en compte l'apport de la lumière naturelle, les capteurs de lumière enregistrent le niveau d'éclairement et permettent une commande automatique des luminaires en fonction de la lumière du jour afin d'obtenir un niveau d'éclairement souhaité, ils permettent aussi de limiter l'exposition des œuvres d'art de musées aux rayonnements pouvant être nuisibles.



Fig. I.5 capteur de lumière [M1]

2.2. Les contrôleurs

2.2.1. Généralités

Les contrôleurs constituent les éléments de base dans une boucle de régulation, ils permettent de lire les entrées provenant des capteurs (température, humidité pression,…), d'exécuter ensuite l'algorithme de régulation pour envoyer les ordres vers les actionneurs qui sont chargés de faire varier la grandeur à contrôler dans le sens désiré.

Dans le domaine de la GTB, les contrôleurs peuvent être regroupés en deux grandes catégories [B3] :

1. *les contrôleurs centralisés :* dans cette catégorie, tous les capteurs et les actionneurs sont reliés à un même système de commande, ceci crée des difficultés au niveau du câblage et aussi au niveau de la robustesse du système vis-à-vis des défauts. Par ailleurs, cette solution exige des systèmes de commande assez puissants afin qu'ils puissent effectuer le calcul nécessaire, tout ceci limite l'utilisation de cette solution.

2. *Les contrôleurs décentralisés :* afin de palier aux limitations des systèmes de commande centralisés, on utilise de plus en plus des systèmes de commande décentralisés ou DCS. Un système DCS (Distributed Contrôle system) est un système de commande caractérisé par la présence de, non pas un seul système de commande centralisé , mais plutôt par la présence de plusieurs systèmes répartis dont chacun commande une entité (sous système) donnée, un bus ou un réseau de communication est ainsi nécessaire pour assurer la liaison entre les différents modules du DCS.

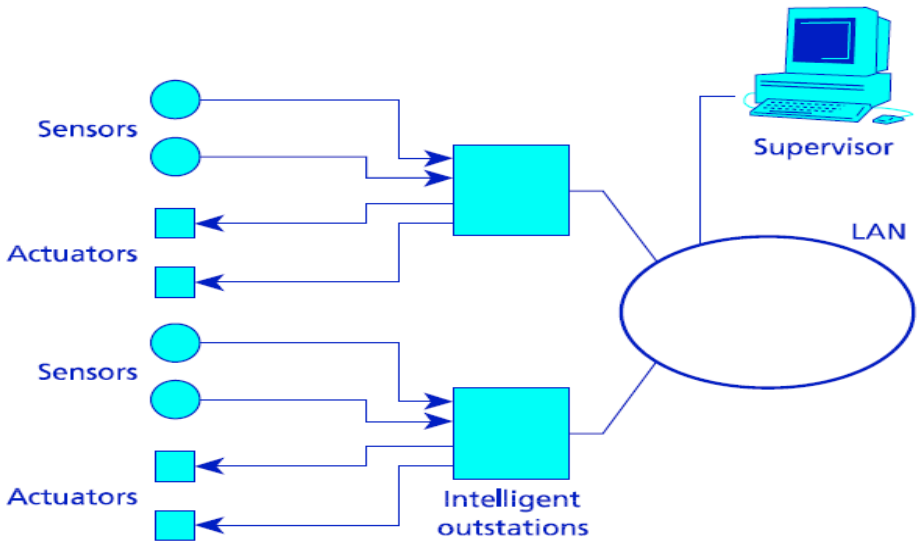


Fig. I.6 système de commande décentralisé [B3]

2.2.2. Description des automates programmables

Les automates programmables commandent les processus aux moyens d'instructions spéciales stockées dans leurs mémoire.

Ils utilisent pour cela, une interface de sortie constituée de connexions auxquels l'automate applique une tension de 24V par exemple ou de 0V :

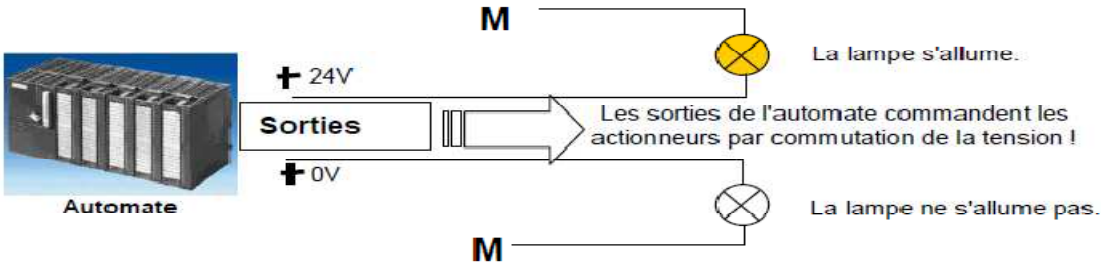


Fig. I.7. Actionneur TOR commandé par automate

Les API pour fonctionner doivent aussi recevoir des informations sur l'état des processus pour un traitement et ceci grâce à l'interface d'entrée qui, comme celle de sortie, est constituée de connexion à laquelle les capteurs (modélisés ici par des contacts à ouverture ou fermeture) appliquent une tension de 0V ou de 24V :

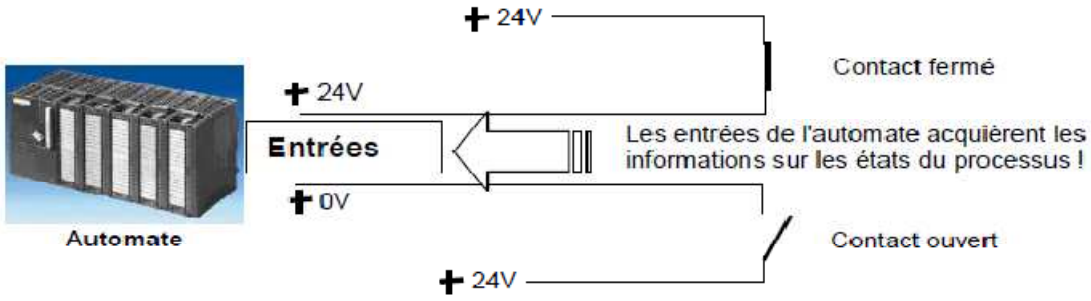


Fig. I.8. Entrées reçues par l'automate

2.2.3. Architecture des API

Le cœur d'automate programmable est sans contestation l'UAL (Unité Arithmétique et Logique) autour de laquelle orbitent d'autres organes.

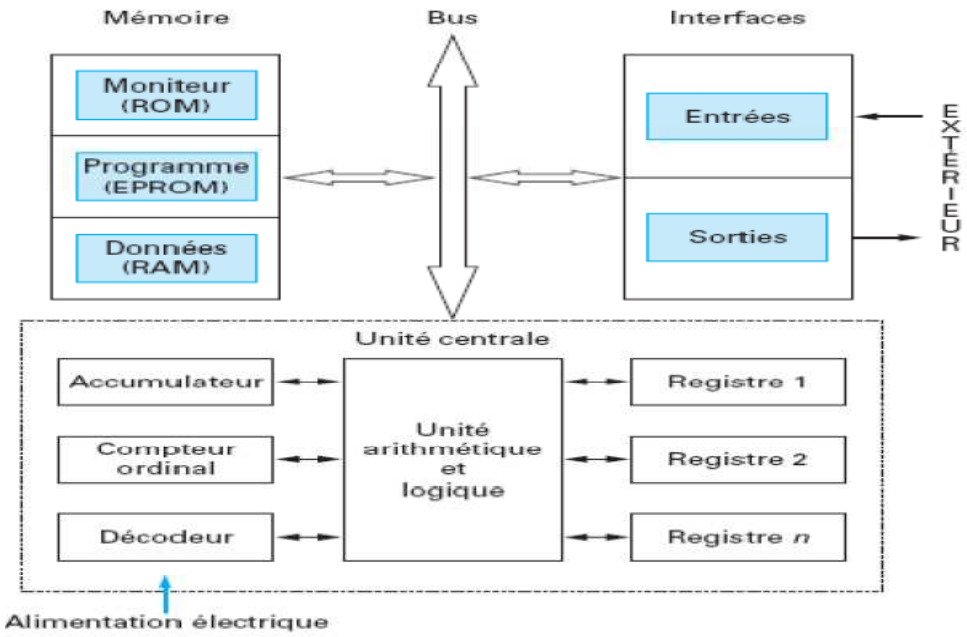


Fig. I.9. Architecture des API

Ces différents organes sont :

module d'Alimentation (P.S) : ce module fournit la tension nécessaire au fonctionnement des API(24V, 15V ou 5V) et ceci en transformant la tension alternative du réseau en courant continu.

La CPU : qui est l'élément central de l'API est constitué du processeur (partie effectuant le traitement) et des mémoires (stockage des données).

Les mémoires : nous avons 4 types de mémoire de base :

- **les ROM ou PROM :** qui sont des mémoires mortes et donc accessibles seulement en lecture, elles servent à stocker les programmes superviseurs.
- **les EPROM :** qui sont des mémoires reprogrammables servant à stocker les programmes utilisateur.
- **les RAM :** qui sont des mémoires vives volatiles mais très rapides servant à stocker les données systèmes et utilisateur lors du fonctionnement des API.
- **Les Mémentos :** sont des éléments bistables servant à mémoriser des états logiques 0 ou 1 et sont utilisés pour les opérations internes des automates pour lesquelles l'émission d'un signal n'est pas nécessaire. On distingue deux types de mémentos: **les mémentos rémanents et non rémanents.**

  Les **mémentos rémanents** conservent leurs états après une panne de la tension de service et ceci grâce à une pile de sauvegarde. Ils conservent aussi leur état en passant du mode RUN au mode STOP, toutefois ils peuvent être remis à zéro soit par l'utilisateur, soit par la commande d'effacement général.

  Les mémentos non rémanents sont remis à zéro lors d'une panne de tension ou lors du changement de mode de RUN à STOP.

Les modules d'E/S : Constituent l'interface entre la CPU et le processus, de ce fait, ils traduisent les signaux industriels en information compréhensible par l'automate et inversement. On distingue trois types de modules E/S : **TOR (Tout Ou Rien), Analogique et des modules spécialisés.**

2.2.4 Traitement du programme dans l'automate

Le traitement du programme dans l'API se fait d'une manière cyclique en respectant les étapes suivantes :

1-Dés la mise sous tension de l'automate, ce dernier enregistre l'état des entrées dans la **MIE**(Mémoire Image des Entrées) en enregistrant un '1' logique si l'entrée et sous tension et un '0' logique si elle ne l'est pas.

2-Le processeur exécute ensuite le programme (situé dans la mémoire programme) séquentiellement en prélevant les entrées requises de la MIE et en écrivant les résultats logiques dans la **MIS** (Mémoire Image des Sorties). Durant l'exécution du programme, le processeur accède aussi aux zones mémoires des mémentos, des compteurs et des temporisations.

3-Dans cette dernière étape, les sorties sont activées ou désactivées suivant l'état de la MIS et le programme reprend au point 1.

On définit le **temps de cycle** comme étant le temps requis par l'automate pour l'exécution du programme.

Les étapes précédentes sont illustrées dans la figure suivante :

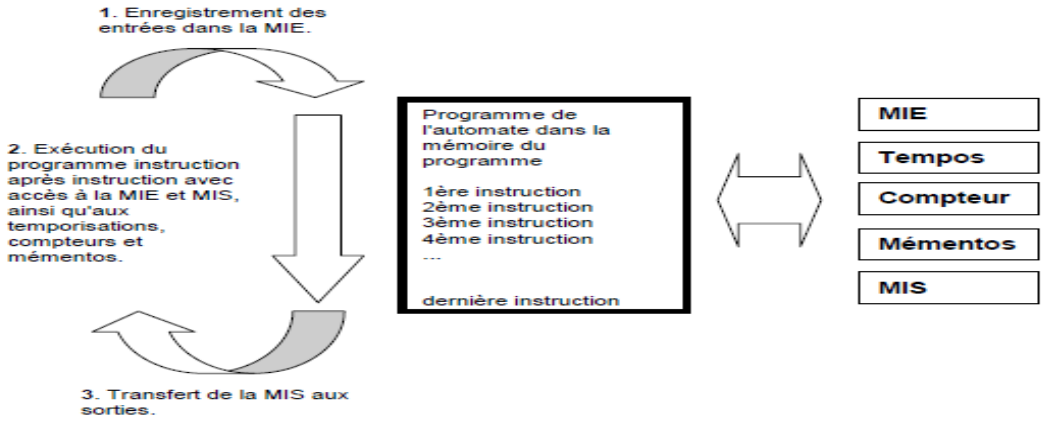


Fig. I.10. Cycle de traitement d'un programme

2.2.5. Les contrôleurs spécialisés

En plus des automates programmables 'classiques', d'autres contrôleurs spécialisés et dédiés au domaine de l'automatisation des bâtiments sont aussi utilisés. Le contrôleur IHC (pour Intelligent Home Control) est l'un des contrôleurs les plus répandus qui sont dédiés au domaine du bâtiment, c'est un **système centralisé** pour la gestion d'une installation électrique, toutes les entrées (bouton-poussoir, détecteurs de mouvement et autres capteurs) et toutes les sorties (lampes, volets-roulants, chauffage, ventilation…) sont ramenées au **coffret électrique**. Le système est construit autour **d'un petit automate** (le contrôleur IHC) qui envoie les ordres d'exécution (aux sorties) selon la programmation réalisée et en fonction des entrées. Ce contrôleur possède un large choix de fonctions : commandes centralisées, scénarios d'éclairage, contrôle de volets roulants …etc.



Fig. I.11. Contrôleur IHC de Schneider [M1]

3. Principales boucles de régulation et fonctions dans les bâtiments intelligents [W2]

3.1. Contrôle de l'éclairage

L'éclairage dans un bâtiment peut être contrôlé soit de façon continue ou digitale. Dans le cas digital, l'éclairage se met en marche dés la détection de la détection d'une présence de personne par exemple, en utilisant un détecteur de mouvements. L'éclairage peut aussi être réglé de façon continue (éclairage interne ou externe).Pour l'éclairage interne, on module ce dernier suivant le niveau de lumière dans la pièce et pour l'éclairage externe suivant le temps qu'il fait dehors.

3.2. Contrôle de l'air ambiant

Les circuits d'aération mixent l'air ambiant de retour et celui de l'extérieur afin de garder l'air à l'intérieur du bâtiment sein et de faire des économies sur l'énergie nécessaire au chauffage ou au refroidissement de l'air.

Des capteurs analogique ou numériques de température peuvent être placés dans les pièces, dans les conduits d'air de retour et d'approvisionnement et parfois à l'extérieur du bâtiment.

Les actionneurs peuvent être placés sur les vannes d'approvisionnement ou de retour d'air, sur les vannes d'eau chaude ou froide ou bien sur les ventilateurs.

Les ventilateurs d'approvisionnement et de retour sont actionnés ou arrêtés suivant les températures, le temps qu'il fait à l'extérieur, la pression à l'intérieur du bâtiment ou une combinaison de tous ces paramètres.

1. *Circuits à Volume d'Air Constant (VAC):* Ce sont les circuits les plus simples mais les moins efficaces. Les ventilateurs dans ce genre de circuits ont une vitesse fixe. Les VAC utilisent à la place l'ouverture et la fermeture des vannes d'approvisionnement en eau chaude ou froide qui alimentent les échangeurs de chaleur pour maintenir une température constante dans un espace donné.

2. *Circuits à Volume d'Air Variable (VAV):* Ces unités sont plus efficaces que les précédents. Parmi les systèmes VAV, on trouve celui qui fournit en air pressurisé des réservoirs dont chacun est, en général, affecté à une pièce ou une aire. La pression à l'intérieur des réservoirs peut être contrôlée en agissant sur la vitesse des ventilateurs (à l'aide de variateurs de vitesse). La quantité d'air à l'intérieur des réservoirs est déterminée suivant les besoins de la pièce. Certains réservoirs VAV ont des échangeurs de chaleur internes. Les vannes d'eau froide ou chaude sont ouvertes ou fermées suivant le niveau nécessaire de chaleur demandée à l'intérieur de la pièce.

3.3. Circuits thermiques

Certains systèmes de climatisation nécessitent un circuit d'approvisionnement en eau chaude et froide pour refroidir ou réchauffer l'air à pulser dans les locaux.

Certains systèmes génèrent eux-mêmes de l'électricité grâce aux turbines à gaz (ou à vapeur) et utilisent l'échappement chaud de ces turbines pour chauffer l'eau du circuit thermique.

1. *Systèmes d'eau froide:* L'eau froide est souvent utilisée pour refroidir l'air des bâtiments et des équipements. Ces systèmes de refroidissement contiennent en général des pompes et des groupes frigorifiques ainsi que des capteurs de température qui mesure la température de l'eau froide.

2. *Systèmes d'eau chaude:* Cette eau chaude peut servir par exemple à l'approvisionnement des VAV à travers les serpentins de chauffage. Ces systèmes contiennent généralement des chaudières et des pompes ainsi que des capteurs de température placés dans les conduites d'approvisionnement et de retour de l'eau chaude.

3.4. Sécurité et Alarmes

Dans un système d'automatisation de bâtiments, si un problème est détecté, une alarme est enclenchée pour avertir par exemple un opérateur. Ces avertissements peuvent être envoyés par email, par téléphone ou être seulement des alarmes audibles et/ou visibles.

Exemples d'alarmes :

- Alarme pour indiquer que le filtre à air est sale.
- Alarmes de défauts d'actionneurs : vannes, pompes…
- Alarmes indiquant que le taux de monoxyde de carbone ou dioxyde de carbone est trop élevé.
- Alarmes d'intrusion.
- Alarmes d'incendies

Dans des sites contenant plusieurs bâtiments, une panne momentanée de puissance peux causer des centaines de milliers d'alarmes. Les alarmes les plus critiques sont prioritaires et peuvent être programmés pour resurgir à certains intervalles de temps jusqu'à ce que ces problèmes soient résolus.

4. Généralités sur les systèmes de Ventilation [W6]

La ventilation est un procédé qui consiste à renouveler l'oxygène dans les locaux afin de maintenir un équilibre de la concentration du gaz carbonique. La ventilation permet également de réduire la chaleur, les odeurs désagréables et l'humidité et d'évacuer les fumées de

cigarette. Dans une usine ou une raffinerie, les systèmes de ventilation évacuent les contaminants dangereux contenus dans l'air.

4.1 Les différents types de ventilation

4.1.1 Ventilation naturelle

Dans la ventilation naturelle, aucun ventilateur n'intervient. L'air se déplace grâce aux différences de pression qui existent entre les façades du bâtiment et grâce à la différence de masse volumique en fonction de sa température.

Des amenées d'air (grilles réglables) doivent être disposées en façade pour les locaux dits "propres" (bureaux, séjours, ...). Des ouvertures de transfert permettent le passage de l'air vers les locaux dits "humides" ou "viciés" (sanitaires, cuisine, ...). Dans ces derniers, l'air est évacué grâce à des conduits verticaux débouchant en toiture.

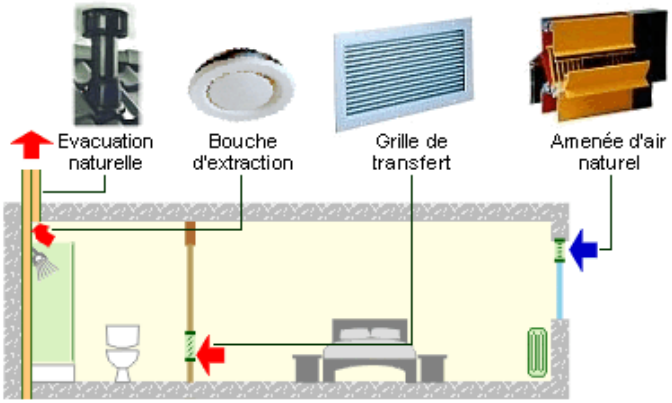


Fig. I.12 Principe d'une ventilation naturelle [W6]

4.1.2 Ventilation simple flux

On parle de ventilation simple flux lorsque soit l'évacuation d'air, soit l'amenée d'air est réalisée grâce à un ventilateur.

La ventilation "simple flux" la plus rencontrée, consiste à créer un mouvement de circulation de l'air dans le bâtiment de telle sorte que l'air neuf entre naturellement par les locaux "propres" (bureaux, chambres d'hôtel,...) et que l'air soit extrait par un ventilateur dans les locaux "humides" (sanitaires, buanderies,...) ou "viciés" (WC, cuisines,...). L'air chemine ainsi à travers plusieurs locaux par ordre croissant de pollution, en passant sous les portes ou par des grilles de transfert.

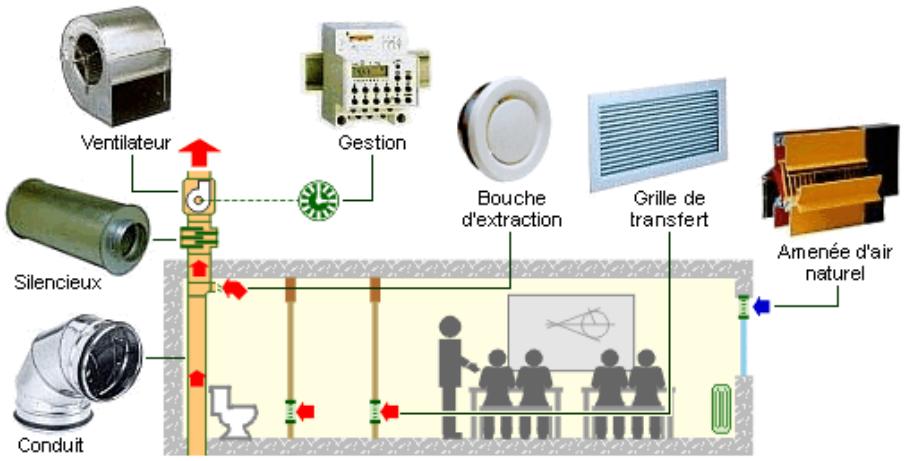


Fig. I.13 Principe d'une ventilation simple flux [W6]

4.1.3 Ventilation double flux

La ventilation "double flux" consiste à organiser :

- la pulsion mécanique d'air neuf, filtré, dans les locaux,
- l'extraction mécanique d'air vicié des locaux.

On peut pulser l'air neuf dans les locaux dits "propres" (bureaux, séjour, ...) et extraire l'air dans les locaux "humides" ou "viciés" (sanitaires, cuisines).

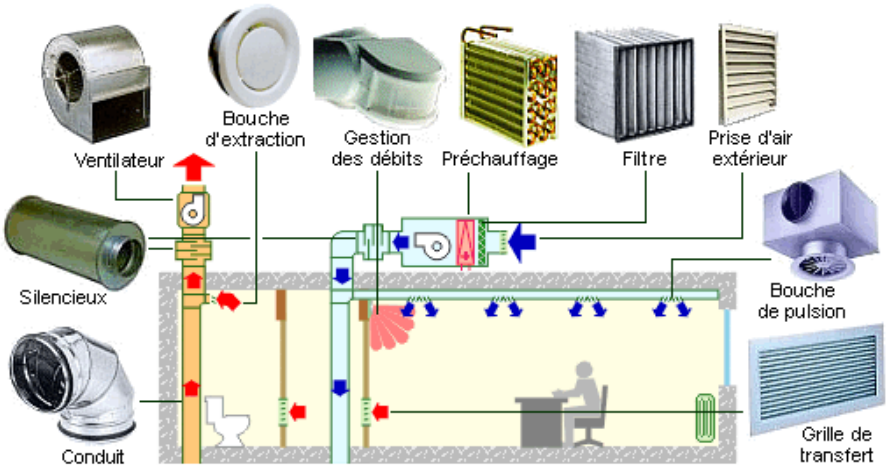


Fig. I.14 Principe d'une ventilation double flux [W6]

4.2 Notions fondamentales sur la ventilation

4.2.1 Notion de perte de charge

Un ventilateur fournit à l'air l'énergie nécessaire pour le déplacer d'un point à un autre (le plus souvent au travers de conduits) en lui imprimant une certaine vitesse.

Le ventilateur fournit donc l'énergie nécessaire à l'air en lui imposant une différence de pression totale (pression statique+dynamique) entre la prise extérieur et la bouche de pulsion (ou dans le sens inverse dans le cas d'une extraction) ; c'est-à-dire, pour mettre l'air en vitesse dans le conduit et vaincre les pertes par frottement dans celui-ci. Cette différence de pression totale est appelée "**hauteur manométrique**" du ventilateur. La perte de pression totale liée à la résistance du réseau de distribution à l'écoulement d'un débit d'air donné est appelée, quant à elle, "**perte de charge**" du réseau.

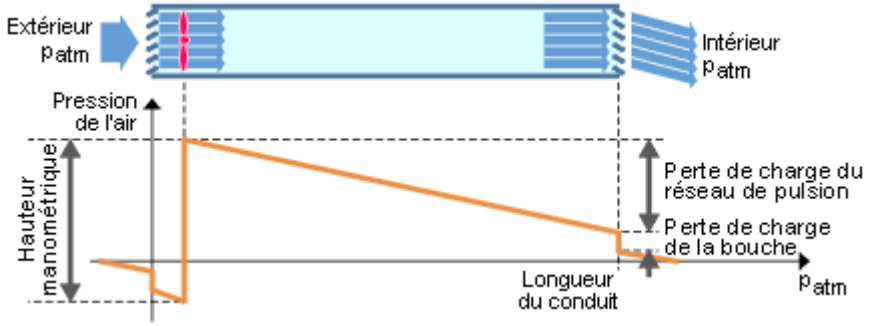


Fig. I.15 Diagramme de pression totale pour un ventilateur [W6]

4.2.2 Courbe caractéristique du réseau de distribution

La résistance du réseau de distribution dépend d'une part de sa configuration (longueur et forme des conduits, changements de direction, obstacles comme les registres, les batteries, les filtres, ...) et d'autre part de la vitesse de l'air qui y circule. En effet, la résistance, ou autrement dit les pertes de charge, représente le frottement de l'air dans les conduits. Ce dernier augmente avec la vitesse de l'air.

Pour chaque type de circuit, on peut ainsi tracer une courbe qui représente la perte de charge en fonction du débit d'air, image de la vitesse.

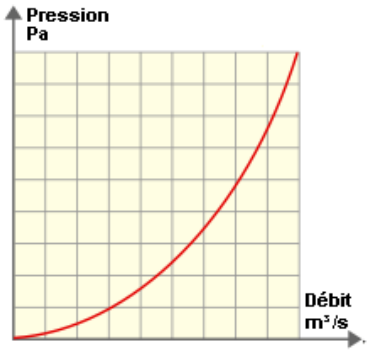


Fig. I.16 Courbe caractéristique du réseau de distribution d'air [W6]

4.2.3 Les courbes caractéristiques d'un ventilateur

Les performances des ventilateurs sont répertoriées sous forme de courbes caractéristiques reprises dans la documentation des fabricants.

On retrouve dans les courbes caractéristiques :

- la hauteur manométrique totale que peut fournir un ventilateur en fonction du débit (ou point de fonctionnement),
- la vitesse du ventilateur pour chaque point de fonctionnement,
- le rendement du ventilateur pour chaque point de fonctionnement,
- la puissance absorbée à l'arbre du moteur.

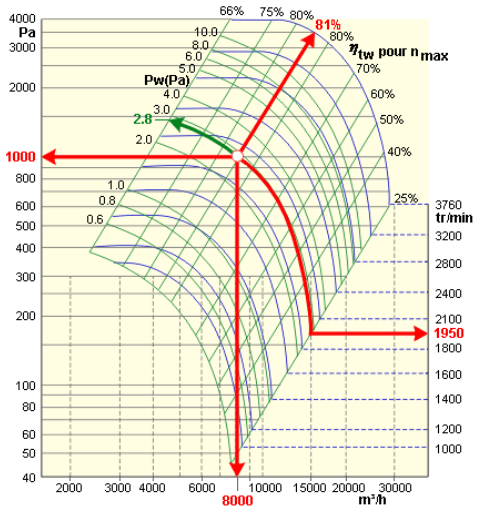


Fig. I.17 Courbes caractéristiques d'un ventilateur centrifuge à aubes inclinées vers l'arrière. [W6]

Dans l'exemple de la **figure VI.6** Pour fournir un débit de 8 000 m³/h, le ventilateur délivre une pression totale de 1 000 Pa. Pour obtenir ce point de fonctionnement le ventilateur doit tourner à 1 950 tr/min. Pour ce point de fonctionnement, son rendement sera de 81 % et la puissance à l'arbre sera proche de 2,8 kW.

4.2.4 Point de fonctionnement

Si l'on branche un ventilateur sur un circuit de ventilation, il stabilisera son débit à une valeur pour laquelle la pression qu'il fournit équivaut à la résistance du circuit. Ce point est le seul point de fonctionnement possible. Il correspond à l'intersection des **courbes caractéristiques du ventilateur** et du circuit. Il définit la **hauteur manométrique** et le débit fournis par le ventilateur lorsque, fonctionnant à une vitesse donnée, il est raccordé au réseau considéré.

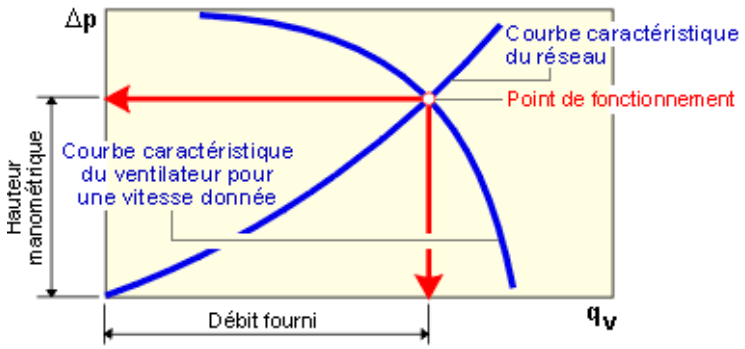


Fig. I.18 Point de fonctionnement d'un ventilateur [W6]

4.2.5 Règles de similitude pour un ventilateur

Ces règles sont résumées dans le tableau suivant :

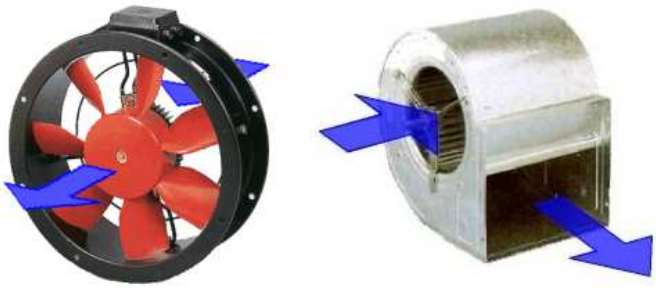
Variation des grandeurs caractéristiques d'un ventilateur lorsqu'on modifie sa vitesse à partir d'un point de fonctionnement donné	
$q_1 / q_2 = n_1 / n_2$	*Légende :*
$p_1 / p_2 = (n_1 / n_2)^2 = (q_1 / q_2)^2$	*q = débit volume (m^3/h)*
$P_{w1} / P_{w2} = (n_1 / n_2)^3 = (q_1 / q_2)^3$	*n = vitesse de rotation (tr/min)*
	p = gain de pression (Pa)
	P_w = puissance sur l'arbre (kW)

Tableau. I.1 Règles de similitude pour un ventilateur [W6]

4.3 types de ventilateurs utilisés

Il existe deux principaux types de ventilateurs :

- **Les ventilateurs axiaux ou hélicoïdes :** l'air est aspiré et propulsé parallèlement à l'axe de rotation du ventilateur.
- **Les ventilateurs radiaux ou centrifuges :** l'air est aspiré parallèlement à l'axe de rotation et propulsé par force centrifuge perpendiculairement à ce même axe. Il existe des ventilateurs à aubes recourbées vers l'avant (à aubages avant) et à aubes recourbées vers l'arrière (à aubages arrière) ou à aubes radiales. Il existe aussi des ventilateurs centrifuges à deux ouïes d'aspiration. Ces roues plus larges, parfois composées de deux roues simple ouïe accolées, aspirent l'air de chaque côté de la roue.



Ventilateur axial. *Ventilateur centrifuge.*

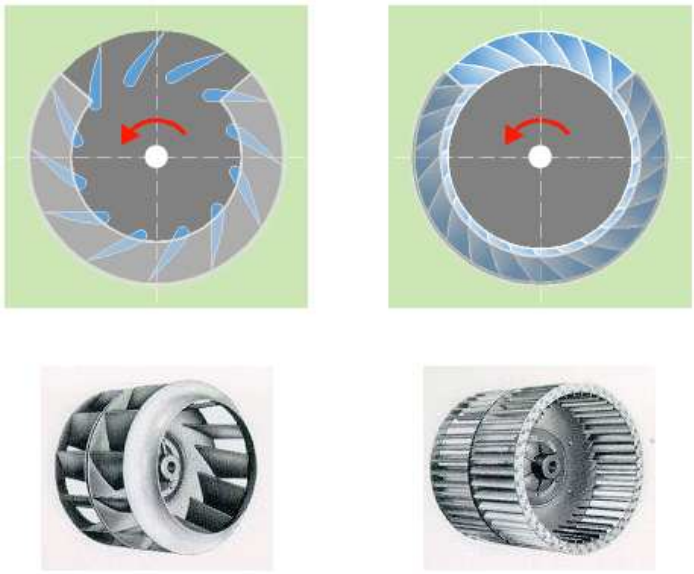
Fig. I.19 Les deux principaux types de ventilateurs [W6]

Etant donné que dans les applications de ventilation ou de climatisation, c'est le ventilateur centrifuge qui est le plus souvent utilisé, c'est ce dernier qui sera explicité par la suite.

4.4 Les ventilateurs centrifuges

Là encore, il existe deux principaux types de ventilateurs centrifuges qui sont :

- **A aubes inclinées vers l'avant** : appelés aussi **"à action"** ou **"en cage d'écureuil"**. La roue de ces ventilateurs comprend un nombre important d'aubes de faible hauteur. Elles sont inclinées dans le sens de rotation de la roue.
- **A aubes inclinées vers l'arrière** : appelés aussi **"à réaction".** La roue de ces ventilateurs comprend un nombre réduit d'aubes de plus grande hauteur. Elles sont inclinées dans le sens inverse de la rotation de la roue.



Roue de ventilateur centrifuge à aubes inclinées vers l'arrière (double ouïe). *Roue de ventilateur centrifuge à aubes inclinées vers l'avant (double ouïe).*

Fig. I.20 Les deux principaux types de ventilateurs centrifuges [W6]

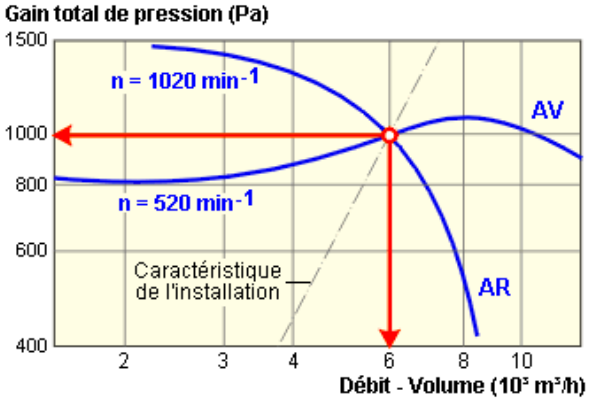


Fig. I.21 Courbe caractéristique d'un ventilateur à aubes recourbées vers l'avant (AV) et à aubes recourbées vers l'arrière (AR), pour un même point de fonctionnement. [W6]

Pour un même point de fonctionnement, la roue avec aubes vers l'avant nécessite toujours une plus faible vitesse de rotation. Le bruit produit par la roue avec aubes vers l'avant sera donc moins intense.

Par contre, lorsque la perte de charge varie dans le circuit aéraulique, la variation de débit qui en résulte est nettement plus importante pour les roues inclinées vers l'avant. Leur point théorique de fonctionnement doit donc être calculé avec beaucoup de rigueur. En effet, si la résistance réelle est inférieure à la résistance calculée, le ventilateur débitera nettement plus que prévu et sa consommation sera supérieure aux prévisions.

A l'inverse des roues à aubes inclinées vers l'arrière, la roue avec aubes vers l'avant ne demande pratiquement pas de variation de vitesse pour obtenir une augmentation importante de débit. (**figure VI.11**)

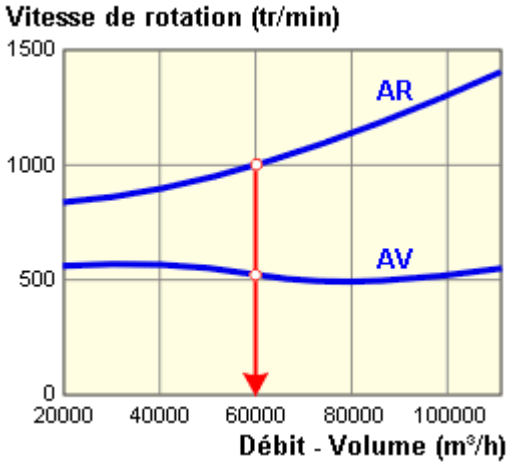


Fig. I.22 Variation du débit d'air en fonction de la vitesse de rotation du ventilateur [W6]

5. Généralités sur les systèmes de climatisation [W6]

La climatisation est la création et le maintien d'un air ambiant dont la température, l'humidité et la pureté sont des facteurs primordiaux. Un système de climatisation est constitué d'un dispositif centralisé, qui produit une atmosphère contrôlée à tout moment, quelles que soient les conditions climatiques.

Dans ce qui suit nous allons décrire quelques systèmes de climatisation utilisés en GTB.

5.1 Système tout air, à débit constant, mono-gaine

Le système de conditionnement d'air "tout air, à débit constant, mono-gaine" est un système où l'air est préparé (chauffé, refroidi, humidifié,...) en centrale dans un caisson de traitement d'air, puis envoyé par un réseau de gaines vers le/les locaux.

On parle de **système tout air** car l'air est le fluide caloporteur de chaleur, de froid, ou d'humidité. Le terme « mono-gaine » quand à lui est du au fait qu'un seul réseau de gaine est crée pour le transport de l'air.

Dans ce type de système, l'ai est traité dans une **CTA** (Centrale de Traitement d'Air) qui a généralement la structure définie par la **figure VI.12**.

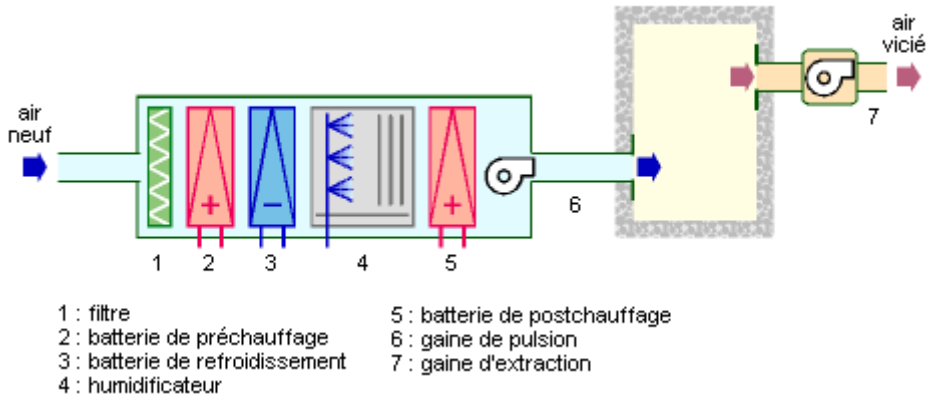


Fig. I.23 Exemple d'une CTA appliquée à un local [W6]

Un exemple d'une configuration d'un bâtiment avec une CTA est donné par la **figure VI.13**.

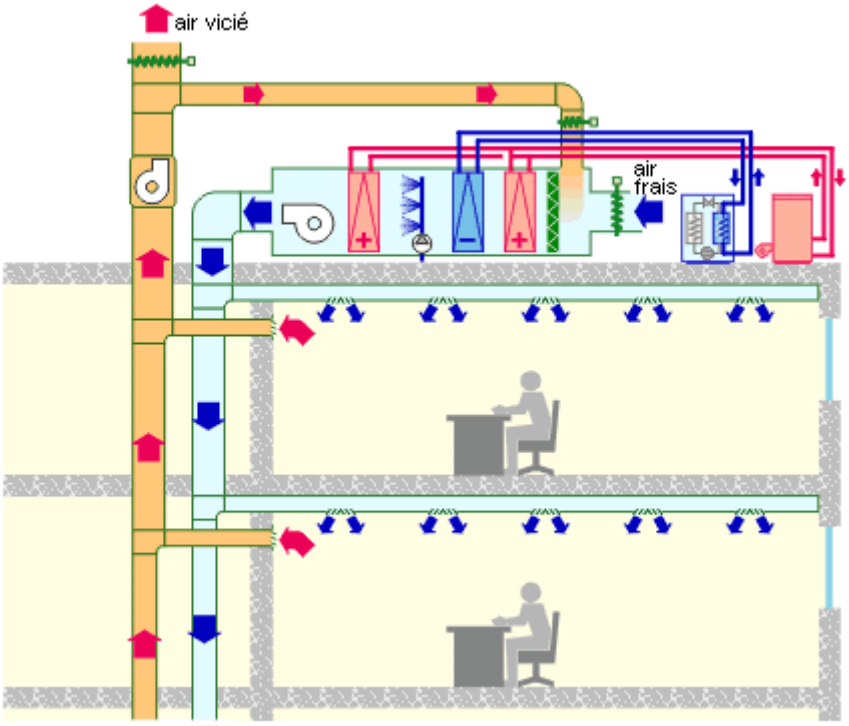


Fig. I.24 Exemple d'une configuration d'un bâtiment avec une CTA [W6]

5.2 Système tout air, à débit variable (VAV)

Le système de conditionnement d'air présenté précédemment délivre un air plus froid lorsque la température extérieure augmente (de 20°, l'air passe à 16°C, par exemple). Le débit d'air pulsé reste le même, mais la température diminue.

Dans un système "tout air-VAV", le débit d'air varie entre le minimum hygiénique pour les occupants et le maximum nécessaire pour reprendre toutes les charges du local (soleil, bureautique, personnes,...).

Une alternative consiste à garder la température de l'air soufflé constante mais à augmenter le débit d'air pulsé au niveau de chaque local. On parle de « système à Débit d'Air Variable » ou «Volume d'air Variable » (VAV).

En pratique, le débit varie entre 30 et 100 % du débit nominal. La variation de débit est faite en agissant :

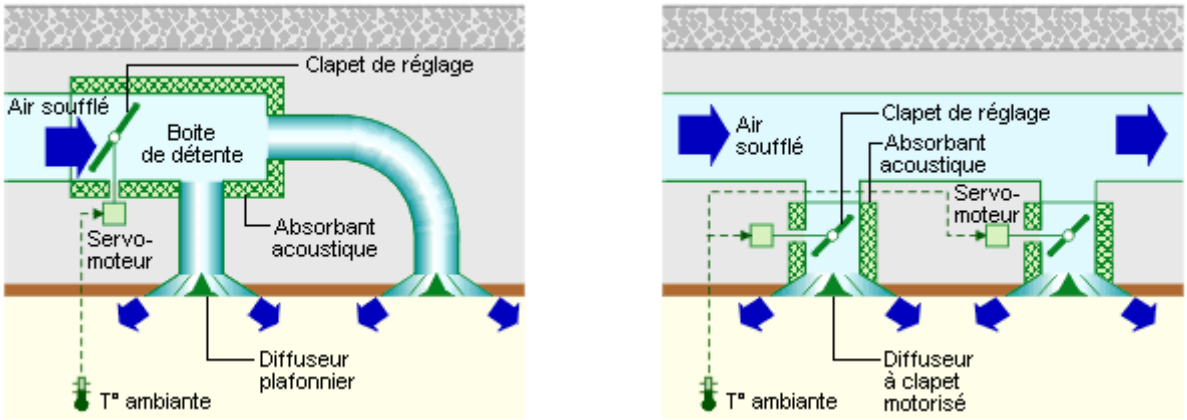
- soit sur un volet motorisé.
- soit directement sur les bouches de soufflage (conçues pour le débit variable).

5.2.1 régulation locale du débit d'air pulsé

On peut adapter le débit par réglage de clapets : un servomoteur commande la position d'un clapet en fonction des besoins du local. Ce clapet est inséré dans une boîte de détente tapissée

d'absorbants acoustiques pour réduire le niveau de bruit (**figure VI.14 -A**). L'air est ensuite réparti vers le local via des diffuseurs.

Il est également possible de faire varier le débit en agissant directement au niveau des diffuseurs. Le clapet est cette fois intégré dans le diffuseur.(**figure VI.14 -B**)



(A) **(B)**

Fig. I.25 Régulation locale du débit d'air pulsé [W6]

Le problème avec les systèmes VAV est qu'une variation du débit d'air entraine une perturbation de pression tout au long du réseau de distribution. En effet, si les bouches se ferment, la pression de gaine va augmente. Toute la distribution de l'air en sera perturbée. Dès lors, on modulera la vitesse des ventilateurs pour maintenir une pression de gaine constante et par la même occasion, la consommation des ventilateurs en sera diminuée.

Si le débit d'air pulsé évolue, il faut que le débit d'air repris évolue conjointement. Il faudra alors agir localement sur le débit des bouches de reprise et on veillera à toujours laisser le local en légère surpression par rapport à l'air extérieur.

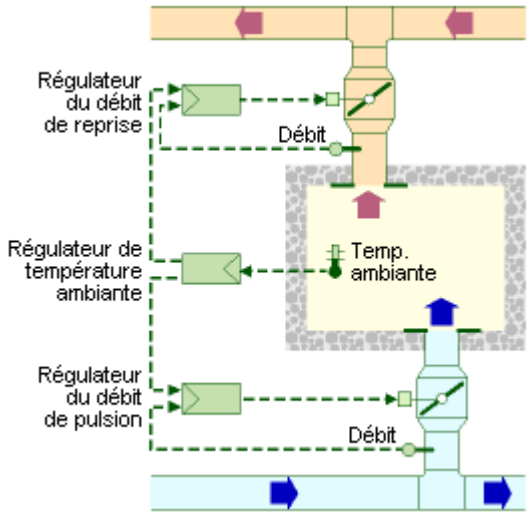


Fig. I.26 Exemple d'une régulation conjointe du débit d'air pulsé et repris [W6]

5.2.2 Régulation globale de la pression dans les conduits d'air

Pour réguler la pression dans un conduit d'air deux solutions existent :

5.2.2.1 Solution 1

Les commandes des ventilateurs de pulsion et de reprise sont synchronisées (le variateur de vitesse agit sur les deux moteurs simultanément). Mais ce système impose que les ventilateurs aient des caractéristiques aérauliques semblables. Or, les deux réseaux sont différents. Des écarts de débit apparaissent et les locaux risquent de ne plus être maintenus en surpression.

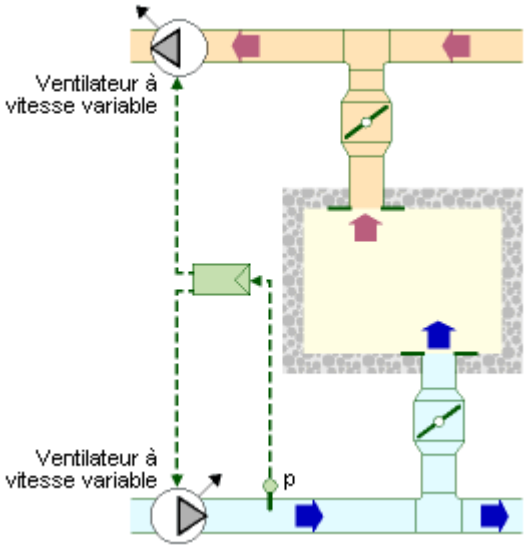


Fig. I.27 Exemple d'une régulation de pression dans un conduit d'air (solution1) [W6]

5.2.2.2 Solution 2

On dispose un capteur de pression dans la gaine de soufflage, un régulateur de pression de gaine reçoit l'information issue de ce capteur, la compare à une référence et commande la vitesse du ventilateur de soufflage.

Pour maintenir une surpression constante au niveau du local, un capteur de pression (ou de débit) est monté dans chacun conduit de soufflage et d'extraction d'air (de préférence prés des ventilateurs). Un deuxième régulateur de pression (ou de débit) commande le ventilateur d'extraction.

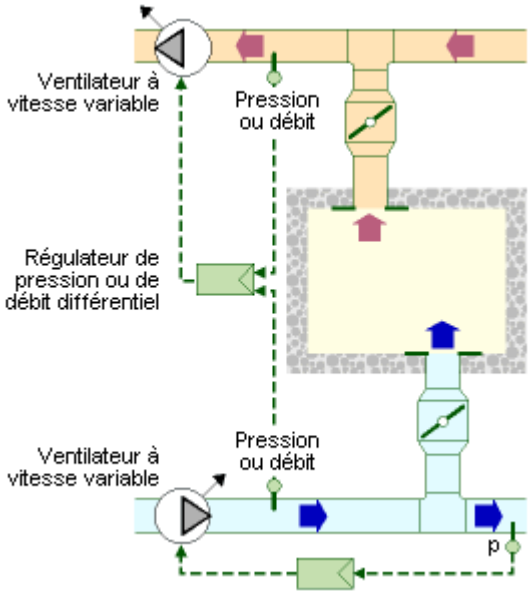


Fig. I.28 Exemple d'une régulation de pression dans un conduit d'air (solution2) [W6]

5.2.3 Exemple d'un système VAV mono-gaine

La **figure VI.18** donne un exemple d'un système VAV contenant une CTA (Centrale de traitement d'air. Cette CTA reçoit de l'air neuf de l'extérieur qui sera mélangé à une partie de l'air extrait (ce qui réduit le volume d'air importé de l'extérieur et par conséquent l'énergie nécessaire à son transport et son traitement).

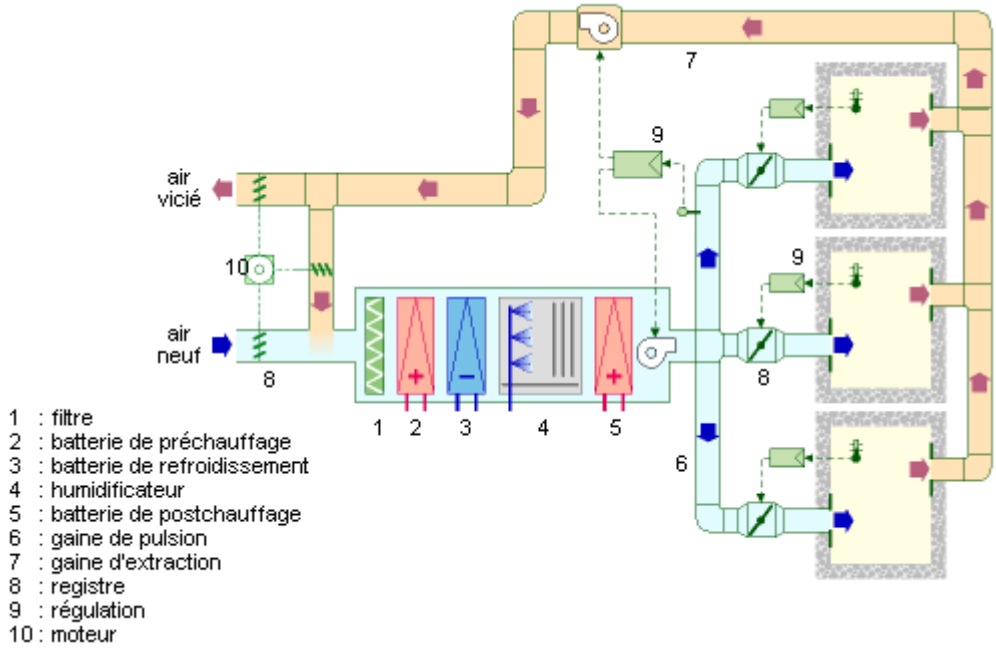


Fig. I.29 Exemple d'un système VAV mono-gaine [W6]

6. Efficacité énergétique

L'économie de l'énergie est devenue un enjeu stratégique à maitriser dans le monde d'aujourd'hui, en effet, une utilisation accrue et non rationnelle de l'énergie conduit à des problèmes très sérieux. D'une part, nous avons le problème de l'épuisement des sources d'énergie devant le fait que le cout des énergies alternatives n'est pas encore maitrisé, et d'autres part, une consommation non rationnelle de l'énergie engendrera aussi le problème de la pollution qui menace le sort de notre planète et des générations futur, or face à cette réalité, on se heurte à un dilemme, en effet :

1. D'une part, on estime que la demande de l'énergie à l'échelle mondiale va doubler dans l'horizon de 2050.
2. D'autre part, et afin de limiter le problème du réchauffement climatique, on estime que les émissions de CO_2 doivent être diminuées de moitié dans l'horizon de 2050 [W4].

Ainsi, afin de résoudre ce dilemme, une utilisation rationnelle et efficace de l'énergie s'impose comme une nécessité incontournable d'autant plus qu'elle est imposée par des législations et des protocoles internationaux (tel que le protocole de Kyoto) [M1]

Devant cette problématique d'économie d'énergie, la GTB s'impose comme une solution de choix, puisque elle permet une utilisation plus efficace de l'énergie d'autant plus que le secteur du bâtiment est l'un des plus grand consommateurs d'énergie et ceci est démontré par les statistiques suivantes : [M3]

- Habitat : 100 à 300 kWh/m².an
- Hôpital : 400 kWh/m².an
- Bureaux : 150 à 350 kWh/m².an
- Data Center : 800 à 1200 kWh/m².an
- Centres commerciaux : 500 à 800 kWh/m².an.

L'efficacité énergétique (ou l'efficience énergétique) permet de réduire les consommations d'énergie à **service rendu égal** entrainant de ce fait la réduction de la facture énergétiques.

6.1. Avantages de l'efficacité énergétique

Dans cette section, on donnera quelques exemples pour illustrer l'importance de l'efficacité énergétique dans la résolution des problèmes liés à la pollution et l'environnement.

Nous savons bien que la consommation de l'énergie est responsable des émissions de gaz qui engendrent des problèmes écologiques tel que les pluies acides et le réchauffement climatique. Pour illustrer l'impact d'une consommation excessive de l'énergie sur l'environnement, considérons l'équation de combustion du méthane (principal composant du gaz naturel)

$CH_4 + 2\ O_2 = CO_2 + 2\ H_2O$

En faisant le bilan de masse on aura

(12 + 4*1) +2(2*16) = (12 + 2*16) + 2(2*1 +16)

(Ce bilan est fait on tenant en compte le nombre de masse du carbone (12) de l'oxygène (16) et de l'hydrogène (1))

Ainsi, la combustion de 16 unité de masse du méthane donne 44 unité de masse du dioxyde de carbone, le rapport est donc de 2.75. Ceci montre donc que la réduction de combustion de méthane grâce à une utilisation efficace de l'énergie permet de réduire considérablement les émissions du dioxyde de carbone et ainsi de limiter le phénomène de réchauffement climatique.

Par ailleurs, sachant que les centrales électriques utilisant le charbon et les fuels sont les principaux responsables des émissions du dioxyde sulfurique responsable à son tour des pluies acides, on déduit que la réduction de consommation de l'énergie apportée par une utilisation efficace de cette dernière permet aussi de réduire les émissions du dioxyde sulfurique et ainsi limiter le problème de pluies acides.

En résumé, l'efficacité énergétique permet de **limiter les problèmes environnementaux** liés à une consommation excessive de l'énergie tout en **réduisant la facture de consommation** pour les utilisateurs [B4].

6.2. Les piliers de l'efficacité énergétique [M3]

L'efficacité énergétique se base sur trois piliers comme le montre la **figure I.30**

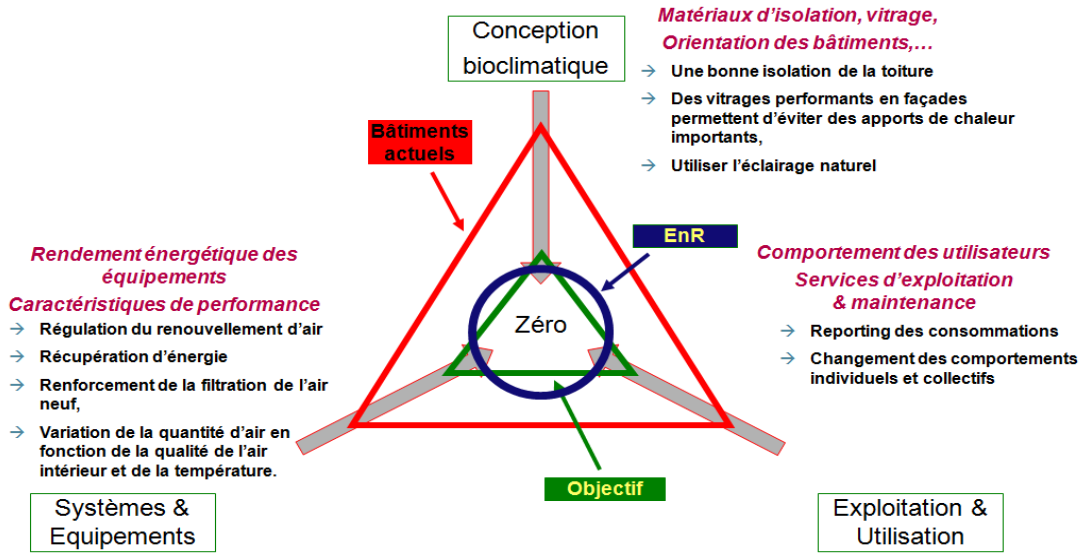


Fig. I.30. Les trois piliers de l'efficacité énergétique [M3]

6.3. Rôle de la GTB dans la promotion de l'efficacité énergétique

Un des exemples illustrant le rôle que joue la GTB dans l'efficacité énergétique est l'exemple d'un système de GTB permettant une commande adaptative de l'éclairage intérieur suivant l'intensité d'éclairage naturelle du soleil. Ainsi on aura un service égal à moindre coût. Par ailleurs, l'utilisation des capteurs de présence permet d'éteindre la lumière là où aucune présence humaine n'est enregistré, ce qui contribue à limiter le gaspillage de l'énergie.

Par ailleurs, les commandes intelligentes des systèmes CVC (Chauffage, Ventilation et Climatisation) permettent de réduire la température des locaux lorsque ceux-ci sont vides tout en garantissant une température agréable à l'arrivée des occupants. Ces commandes permettent aussi d'agir sur les stores afin de profiter de la chaleur du soleil en hiver et de s'en protéger en été, la facture de chauffage et de climatisation se trouve ainsi nettement réduite. Ceci démontre donc le rôle capital que joue le GTB dans la résolution d'un enjeu capital qui est l'efficacité énergétique [B4].

6.4. Empreinte de Schneider Electric dans le domaine de l'efficacité énergétique

A travers plusieurs décennies d'expérience dans le domaine de la GTB, Schneider Electric est considérée aujourd'hui comme un leader mondial dans le domaine des bâtiments intelligents. En effet, Schneider Electric a apporté plusieurs solutions permettant la réduction de la facture énergétique et les a implémentées dans ses propres locaux à l'instar de son nouveau siège situé en France.



Fig. I.31. Vitrine technologique internationale de Schneider Electric [M3]

Un des systèmes développés par Schneider permettant la réduction de la consommation énergétique est le système DALI (Digital Addressable Lighting Interface). En effet, ce système permet de profiter de l'éclairage extérieure en modulant l'éclairage intérieur et en agissant sur les stores. Ce système possède plusieurs caractéristiques: [M3]

- Système **intelligent** de contrôle de l'éclairage permettant la programmation de scénarios d'éclairage.
- Utilisation de réseaux de communication performants et dédiés à la commande de l'éclairage.
- Système Associé aux **capteurs de lumière** et aux **détecteurs de présence**
- **Simplification** des installations et une gestion optimisée des opérations d'exploitation et de maintenance:
    - Scénarios et niveau d'éclairement
    - Adjonction de nouveaux luminaires sans limitation

6.4.1. Présentation de l'application HED (Hive Energy Display)

Toujours dans le cadre de la promotion de l'efficacité énergétique, Schneider Electric a développé une application permettant d'afficher en temps réel la consommation énergétique dans le but d'une meilleure gestion de cette dernière. Cette application a été implémentée au niveau du siège générale de la société en France (the Hive building), ainsi, cette application permet de rendre la consommation d'énergie visible par toute personne à l'intérieur du bâtiment ce qui contribue à mieux sensibiliser les employés sur l'enjeu énergétique.

Cette application dispose de plusieurs fonctionnalités, on peut citer par exemple :

1. *Affichage de la consommation globale cumulative depuis janvier 2009* : et ceci comme le montre la figure suivante :



Fig. I.32. Consommation globale depuis 2009 [M3]

2. *Performance quotidienne :*

Cette fonctionnalité permet de visualiser grâce à des graphiques la consommation d'énergie de la journée précédente comme le montre la figure suivante :



Fig. I.33. rapport quotidien de la consommation de l'énergie par heure [M3]

3. *Pourcentage d'utilisation :* L'application HED permet d'avoir accès au niveau de consommation d'énergie de différents secteurs : l'éclairage, le CVC , les technologies de l'information. Cette fonctionnalité est représentée par la figure suivante :



Fig. I.34. Pourcentage d'utilisation de l'énergie[M3]

4. *Tendances mensuelles :* Cette fonctionnalité permet de visualiser la consommation mensuelle d'énergie et ceci dans le but de la comparer à une courbe représentant les objectifs prédéfinis à atteindre en terme d'énergie. Cette fonctionnalité est illustrée par la figure suivante :

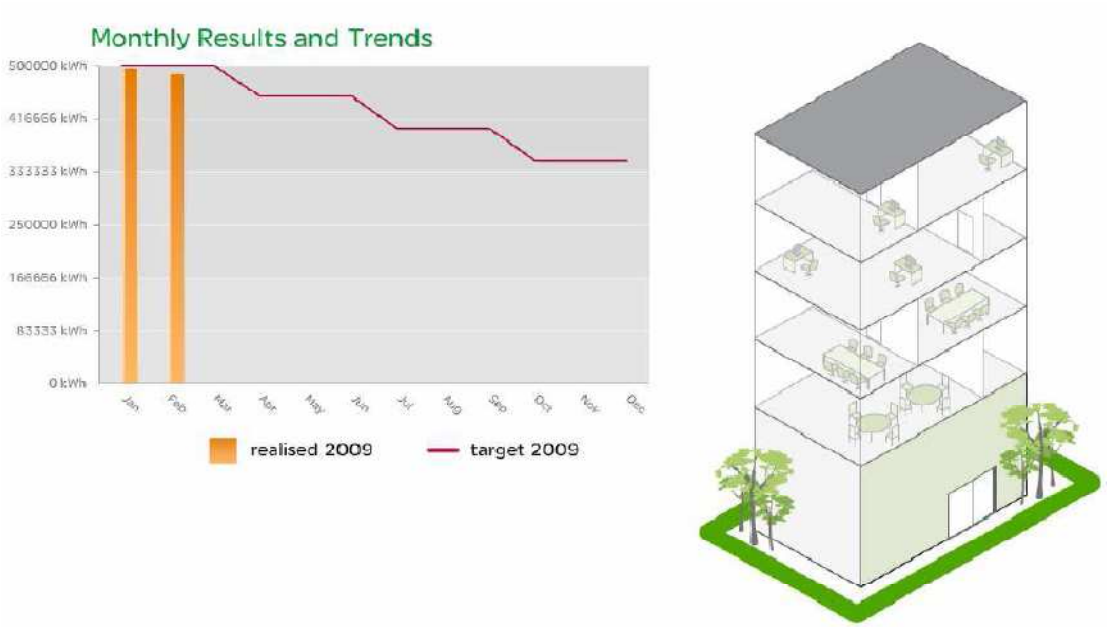


Fig. I.35. Tendances mensuelles[M3]

5. *L'épargne :* Cette fonctionnalité permet de comparer la consommation d'énergie du nouveau siège à celle de l'ancien siège. Les économies faites sont données en : KWH, en euros, en tonnes de CO_2 non émis dans l'atmosphère et en nombre d'hectares de forets sauvés. Cette fonctionnalité est illustrée par la figure suivante :



Fig. I.36. Economies réalisés [M3]

6. *Consommation par étage* : cette fonctionnalité permet de visualiser la consommation d'énergie par étage et pour chaque service dans l'étage (CVC, éclairage…etc.) comme le montre la figure suivante :

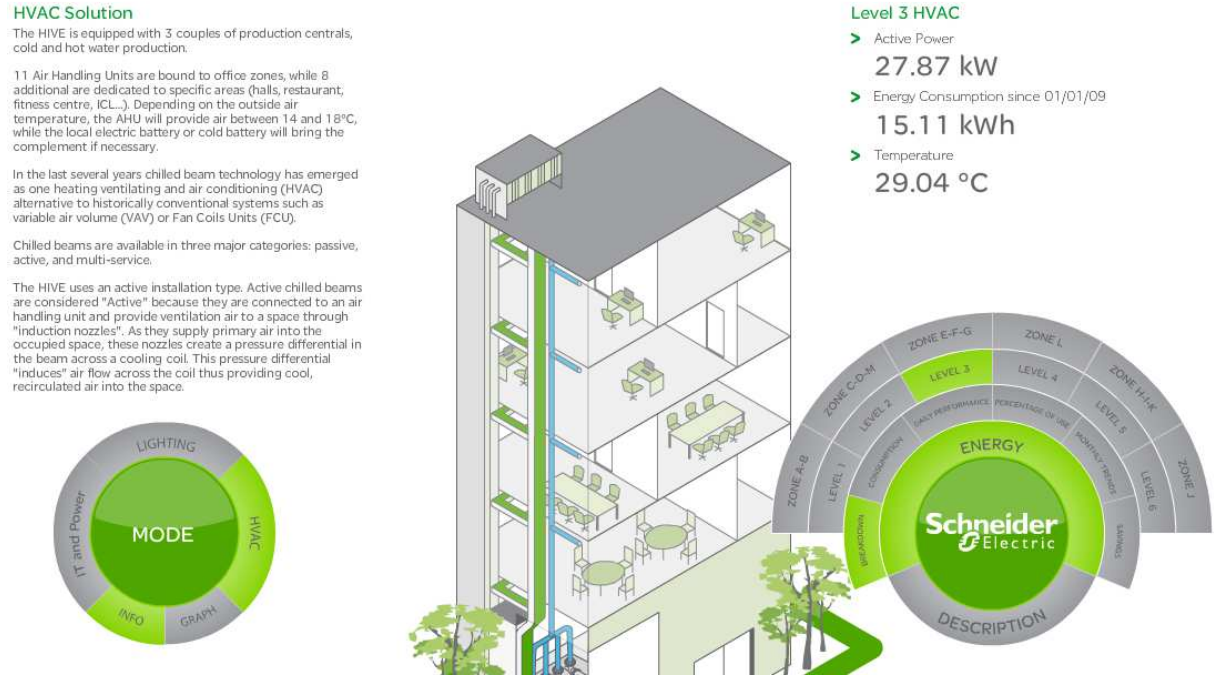


Fig. I.37. Consommation d'énergie par étage et par service [M3]

7. Conclusion

Durant ce chapitre, nous avons donné les définitions générales relatives aux bâtiments intelligents puis introduit les notions de base concernant la GTC/GTB en explicitant les principaux équipements utilisés, nous avons par la suite mis en valeur l'intérêt de tels systèmes en particulier du point de vue énergétique.

Nous avons aussi présenté les principales techniques de ventilation et de climatisation rencontrées dans le monde de la GTB et ceci au travers d'exemples très représentatifs.

CHAPITRE II

Présentation du logiciel Unity Pro

1. Introduction

Unity Pro est le logiciel de programmation des gammes d'automates : Modicon M340, Premium, Atrium et Quantum. Il supporte les cinq langages de la norme IEC 6113-3 qui sont : langage à blocs fonction (FBD), langage à contacts (LD), diagramme fonctionnel en séquence (SFC), liste d'instructions (IL) et littéral structuré (ST).

Dans ce chapitre, nous allons présenter ce logiciel en décrivant ses diverses fonctionnalités.

2. Présentation de l'interface utilisateur d'Unity Pro

En activant le logiciel Unity Pro et après avoir configuré le matériel (chose que nous verrons par la suite), l'interface utilisateur d'Unity Pro apparait comme ceci :

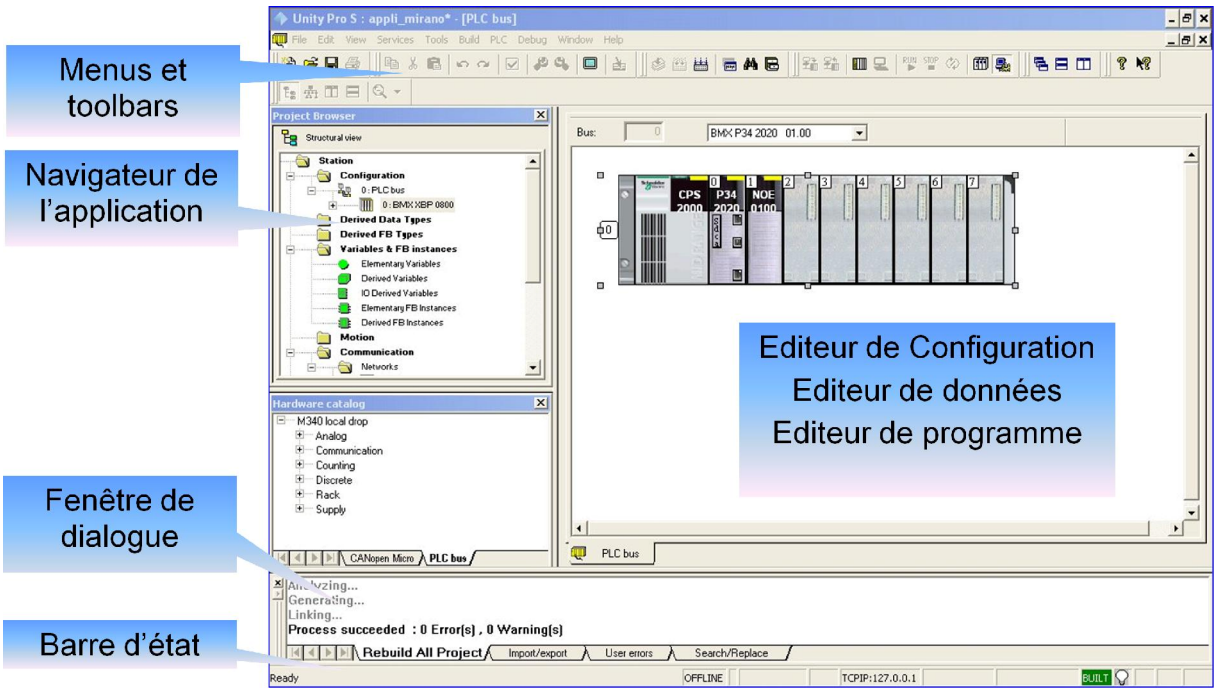


Fig. II.1 Interface utilisateur

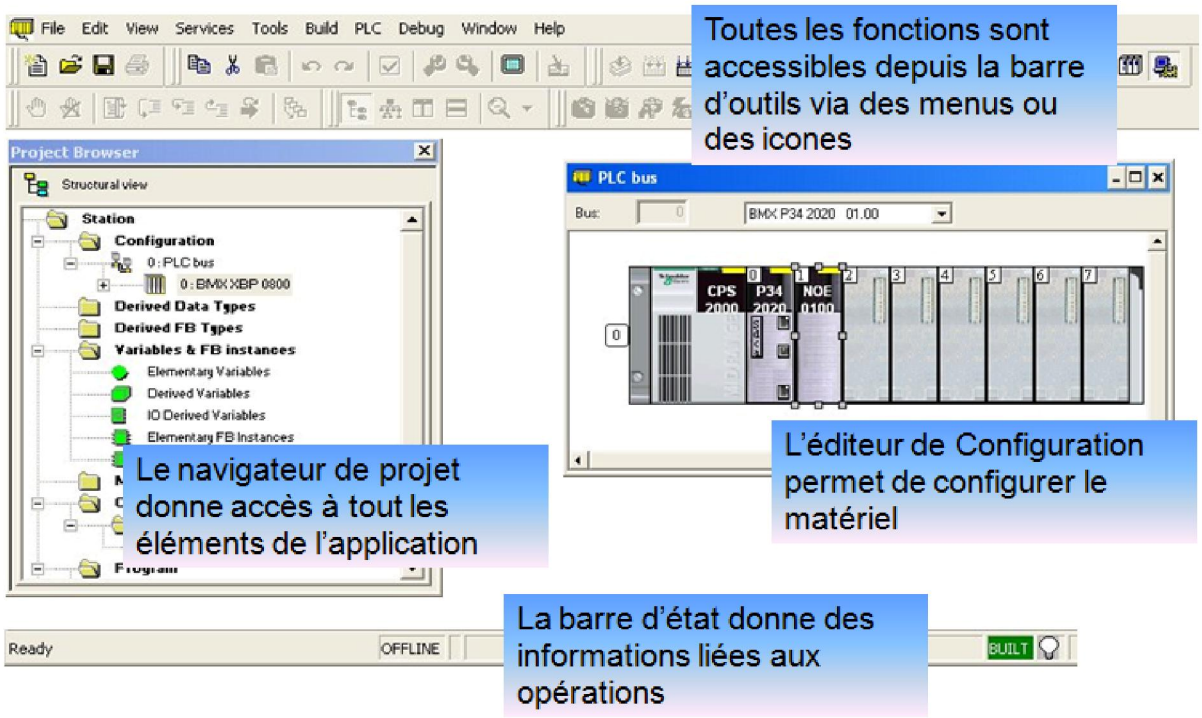


Fig. II.2 Interface utilisateur (suite)

Le navigateur de projet permet de structurer ce dernier sous une forme arborescente afin de faciliter l'accès aux diverses fonctionnalités d'Unity Pro. Ce navigateur propose deux différentes vues, pour structurer et présenter le projet, qui sont : une vue structurelle et une vue fonctionnelle comme le montre la **figure II.3**

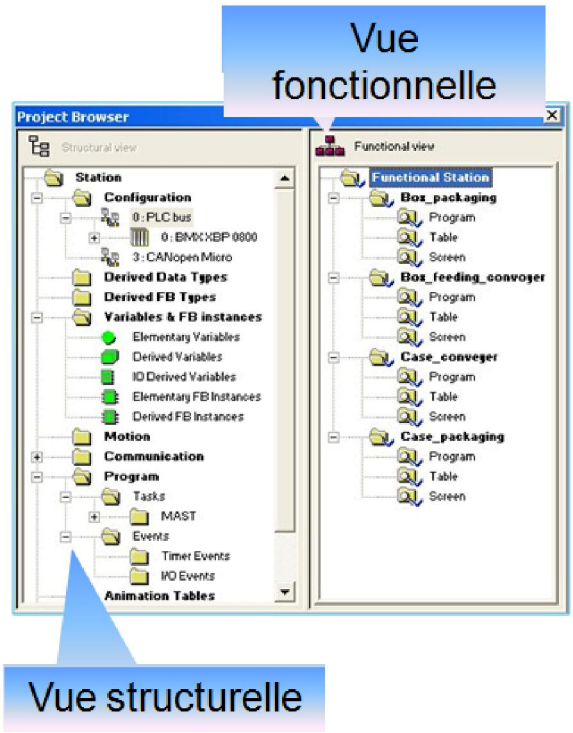


Fig. II.3 les différentes vues du navigateur de projet

Avec la vue structurelle, l'utilisateur peut accéder et gérer les différents éléments de l'application (Configuration matérielle, variables, programmes…) comme le montre la **figure II.4.**

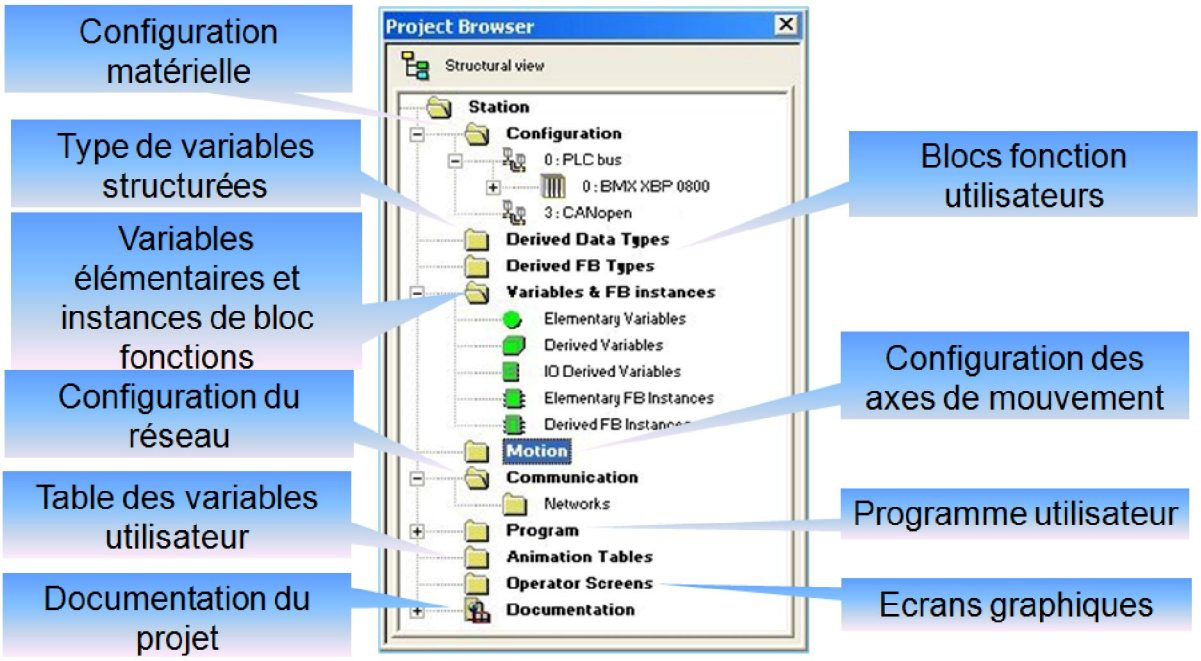


Fig. II.4 vue structurelle du navigateur de projet

La vue fonctionnelle quand à elle permet à l'utilisateur de structurer son projet sous forme de modules fonctionnels comme le montre la **figure II.5**

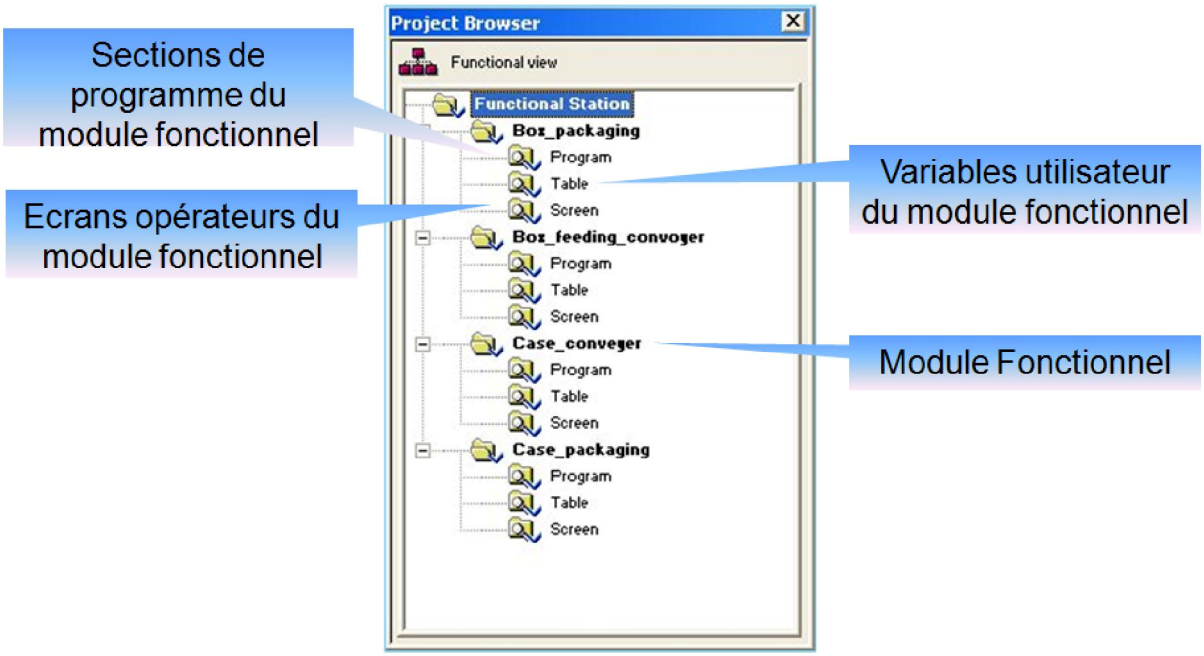


Fig. II.5 vue fonctionnelle du navigateur de projet

3. Création d'un nouveau projet sous Unity Pro

La création d'un projet sous Unity Pro passe par plusieurs étapes afin de faciliter la tâche d'automatisation. Ces étapes sont :

1) Configuration matérielle.

2) Configuration du réseau.

3) Définition et édition des variables (variables élémentaires, structures, tableaux…)

4) Structurer l'application sous forme de tâches, de sections et de modules fonctionnels.

5) Edition du programme dans les cinq langages de programmation mentionnés précédemment.

6) Compilation et simulation des programmes avant leur implémentation dans l'automate.

3.1. Configuration matérielle

Dans cette étape l'utilisateur peut choisir l'unité de traitement qui correspond à ses exigences, ainsi que le rack ou sera placé l'automate, et la configuration des divers modules d'entrées/sorties en fonction du nombre de capteurs et d'actionneurs et de leurs types.

3.1.1. Choix d'une CPU

Dès la création d'un nouveau projet, il est demandé de choisir une station d'automatisation (Modicon M340, Premium, Quantum, Quantum Safety) suivi du choix d'une CPU comme le montre la **figure II.6**

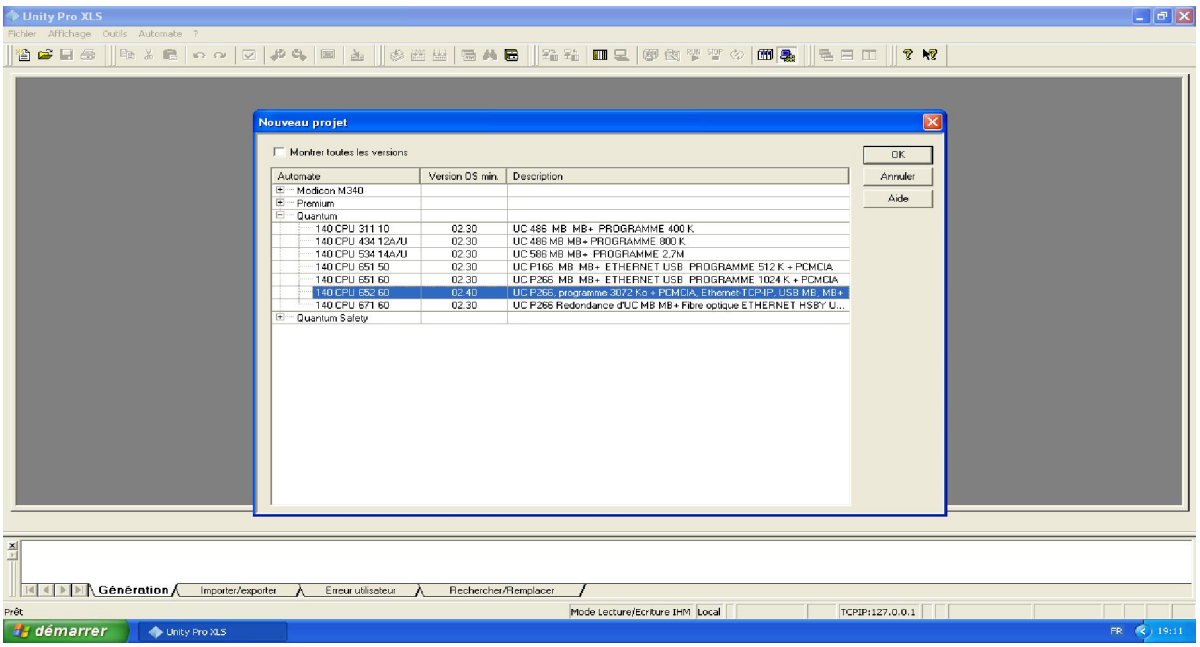


Fig. II.6 choix d'une CPU

Le choix d'une CPU n'est pas irréversible, on peut ainsi remplacer une CPU déjà existante par une autre qui doit appartenir à la même station d'automate que la précédente mais cette opération ne peut se faire que si l'automate n'est pas connecté (mode local). Si l'emplacement de la CPU est incorrect ou bien certains modules d'entrée/sortie ne sont pas pris en compte par la CPU, un message d'erreur est généré.

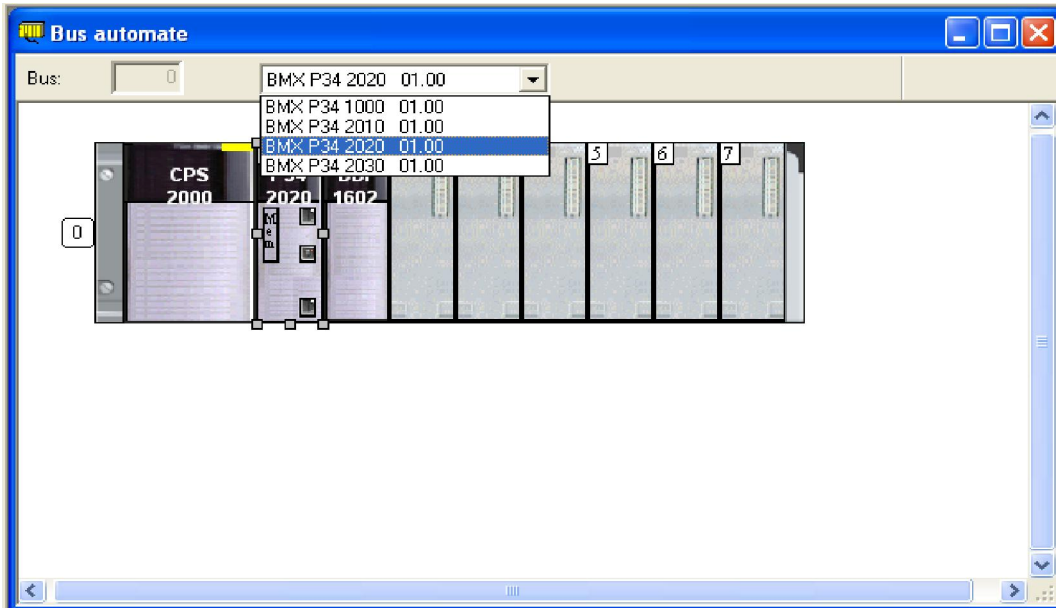


Fig. II.7 remplacement d'une CPU

3.1.2. Configuration d'un rack

La configuration matérielle permet de choisir entre plusieurs types de racks différents et de les adresser. Dès la création d'un projet, un rack est sélectionné par défaut et possédant l'adresse suivante :

- 0 pour un automate de la famille Premium/Atrium ou Modicon M340.
- 1 pour un automate de la famille Quantum.

Le nombre de racks supporté par une station d'automatisation dépend de la nature de la CPU.

Pour ajouter un rack, il suffit de faire glisser le rack déjà existant vers les zones 1 ou 2 comme le montre la **figure II.8.**

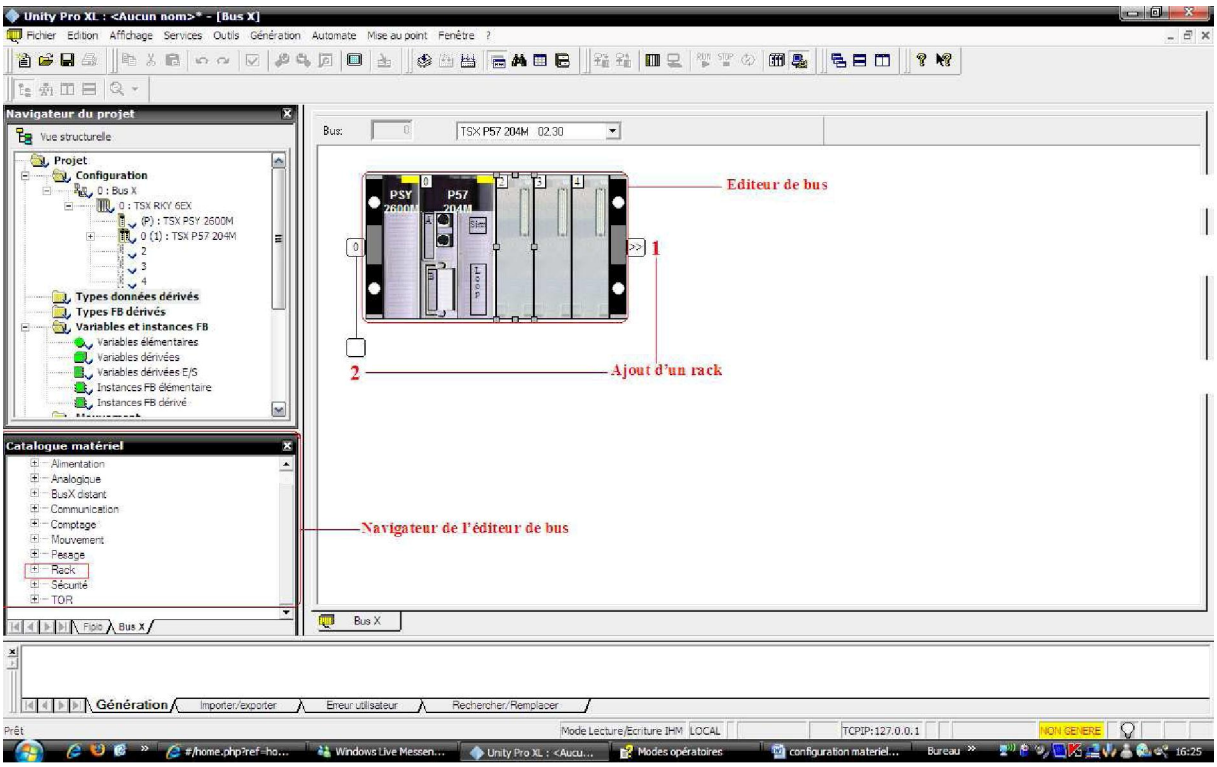


Fig. II.8 Ajout d'un rack

- Si le rack est ajouté vers la direction 1, Il aura la même adresse (ici 0) et les modules seront numérotés à partir de la première puissance de 2 qui vient après le numéro du dernier module.
- Si par contre, le rack est déplacé vers la position 2, le rack aura une adresse différente et la numérotation des modules commencera à partir de l'adresse 0.

3.1.3. Choix du module d'alimentation

Le choix d'un module d'alimentation dépend de plusieurs paramètres parmi lesquels figure le nombre et le type de modules d'entrée/sortie à alimenter. Le choix du module d'alimentation se fait à partir du catalogue matériel comme le montre la **figure II.9**

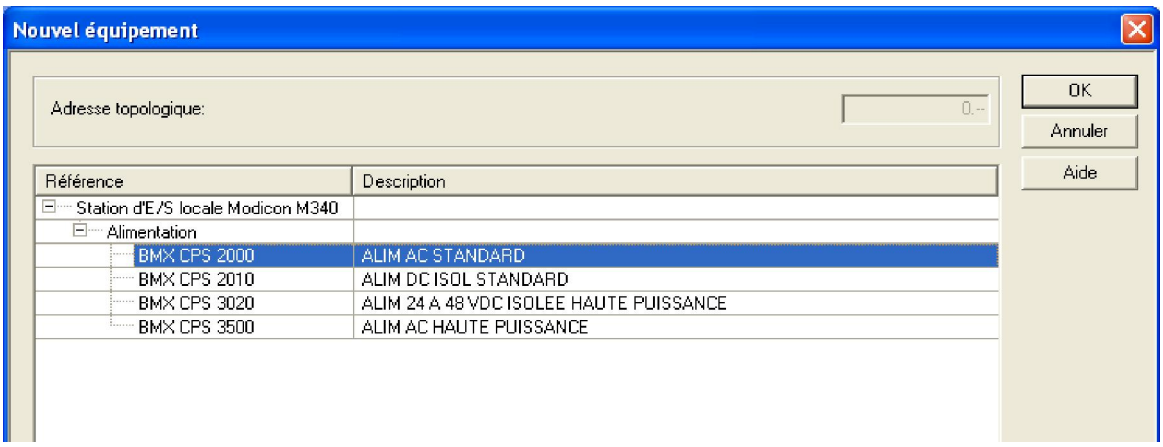


Fig. II.9 choix d'une alimentation

3.1.4. Choix de modules d'Entrée/sortie, de communication et de modules métiers

Le catalogue matériel permet à l'utilisateur de choisir d'autres modules à insérer dans le rack. Parmi ces modules il y a : les entrées/sorties TOR ou analogique, les modules de communication et les coupleurs ainsi que les modules métier (comptage, commande d'axes, commande pas à pas, régulation...)

Pour choisir par exemple un module analogique particulier (suivant sa référence unique) il suffit de déployer l'ensemble des références disponibles en cliquant sur le petit icône « + » puis en double cliquant sur le module recherché comme le montre la **figure II.10**

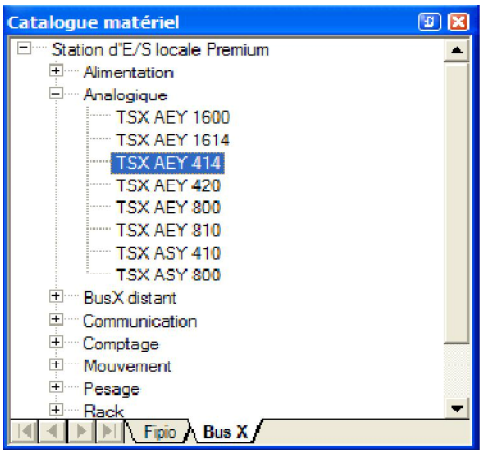


Fig. II.10 exemple de choix d'un module analogique

En double cliquant sur le module choisi dans l'éditeur de configuration matérielle, une fenêtre s'ouvre donnant une description du module.

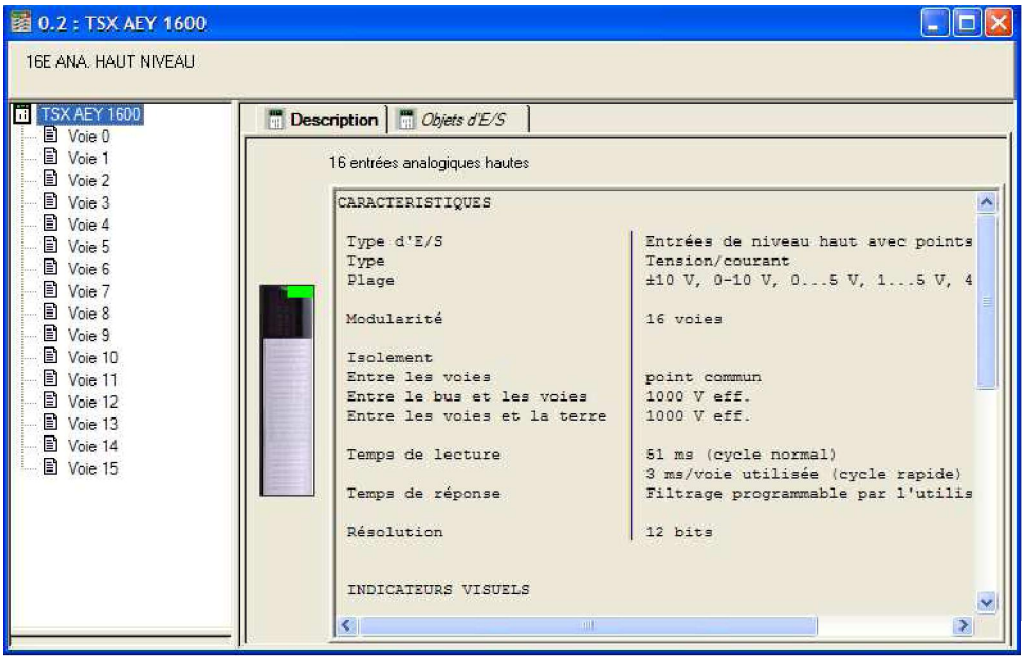


Fig. II.11 fenêtre décrivant le module d'entrée/sortie choisi

Pour un module d'entrée/sortie l'onglet « Objet d'E/S » sur la figure précédente permet de configurer le module de la façon suivante :

- Choix d'un nom pour chacune des voies ce qui implique la création automatique d'une variable élémentaire dans la table des variables.
- Choix du type de la voie : entrée/sortie tension (0-10V, 0-5V…) ou courant (4-20 mA, 0-20 mA…)
- Mise à l'échelle de la voie.
- Filtrage de la voie.

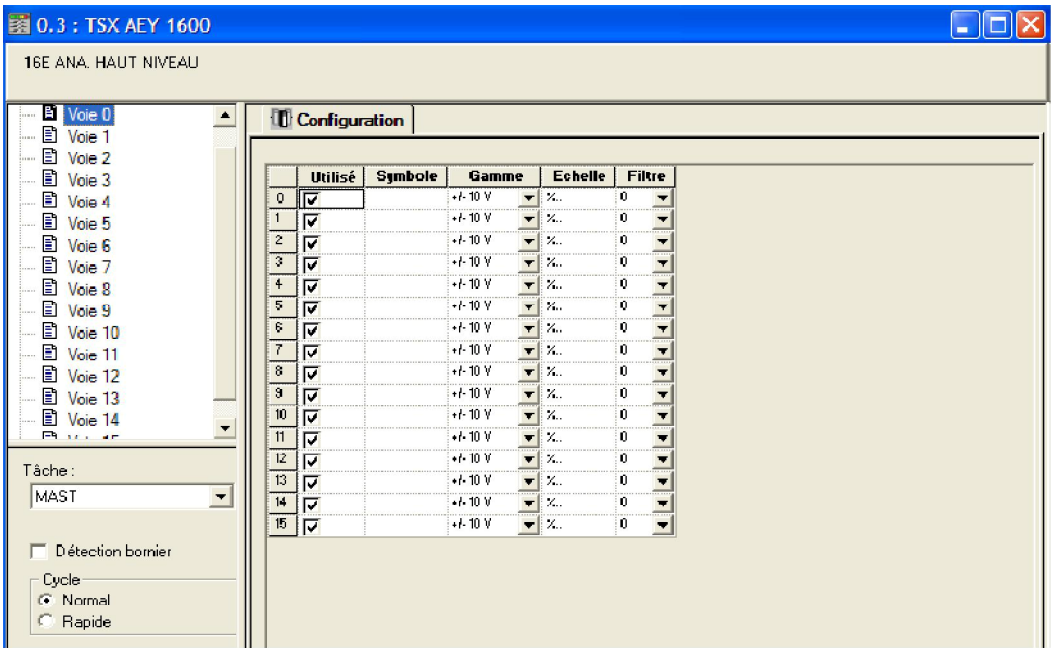


Fig. II.12 exemple de configuration d'une voie

3.2. Configuration d'un réseau

Dans la plupart des applications d'automatisation, la définition et la configuration d'un réseau de communication devient de plus en plus inévitable du fait du développement exponentiel de la technologie des réseaux et son utilisation dans le milieu industriel ou dans notre cas pour automatiser un bâtiment.

- En premier lieu, il faudra créer un réseau logique et lui donner un nom. Cette étape s'effectue en sélectionnant « communication » dans le navigateur de projet puis en cliquant avec le bouton droit de la souris sur « réseaux » et en sélectionnant « nouveau réseau » comme le montre la **figure II.13.**

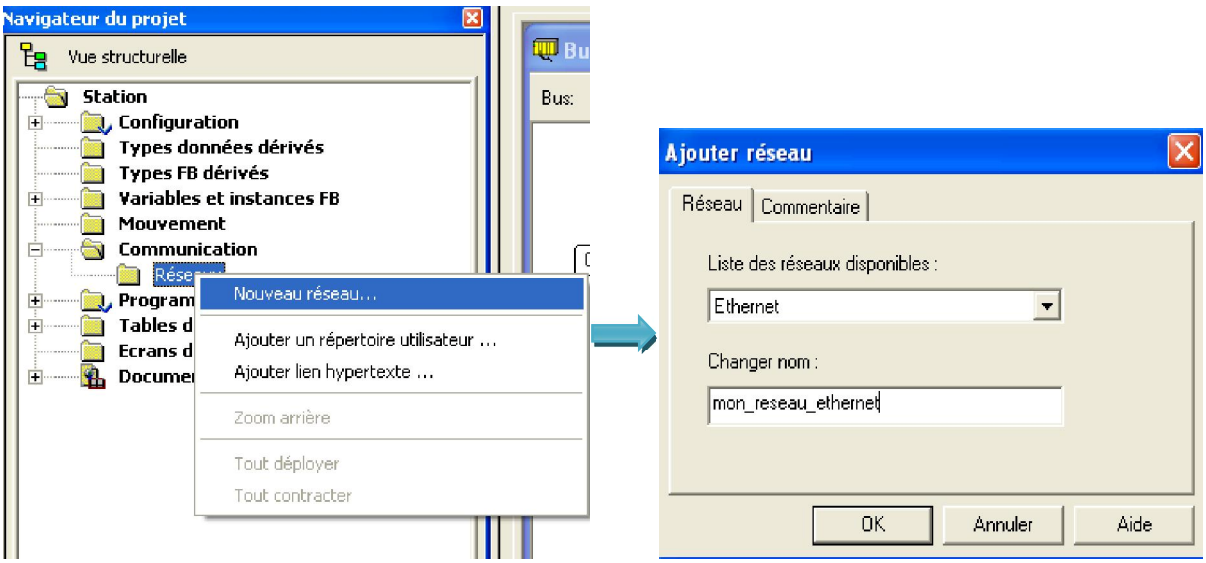


Fig. II.13 création d'un nouveau réseau logique

- On choisit ensuite le type de réseaux dans la liste des réseaux et on lui donne un nom significatif.
- On double clique ensuite sur le nom du réseau logique crée. Une nouvelle page apparait qui va nous permettre de configurer ce réseau logique (adresse IP, type de module de communication…)

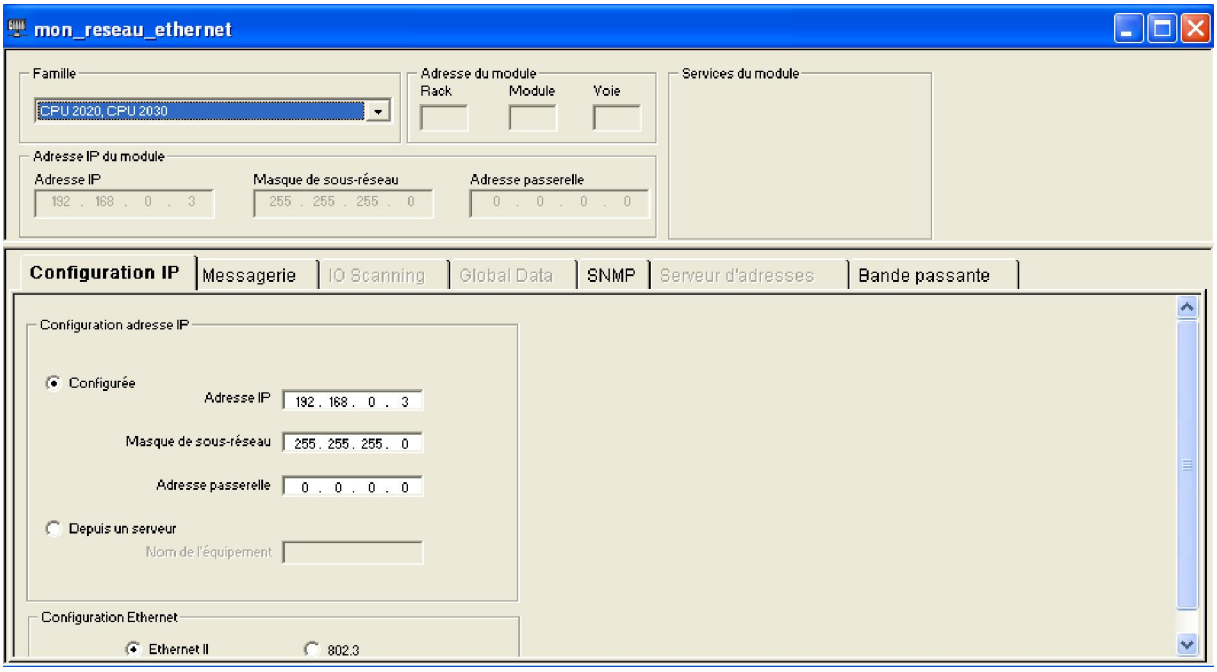


Fig. II.14 configuration du réseau logique

- Après avoir crée le réseau logique, il faut l'associer soit à un module de communication soit à la CPU. Si le réseau est associé à un module de communication, il faut ouvrir le module puis sélectionner la voie sur laquelle sera branché le câble

réseau et lui affecter le nom du réseau logique crée (dans le répertoire **Fonction**). Un exemple est donné par la **figure II.15.**

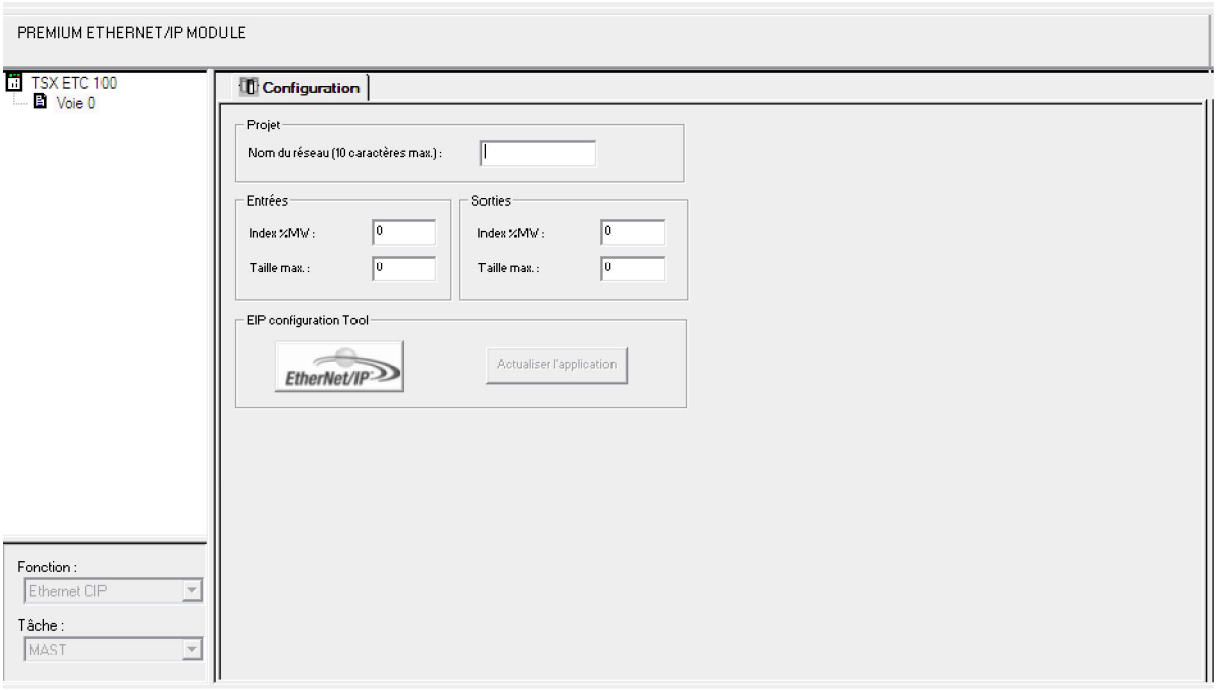


Fig. II.15 association d'un réseau logique à un module de communication

3.3. Définition et édition des variables

3.3.1. Définitions

Une **variable** est une entité mémoire qui peut être modifié par le programme lors de son exécution. Elle peut être de différents types : BOOL (booléenne), WORD (mot de 16 bits), DWORD (mot de 32 bits)….

Une variable peut être **affectée** ou non. Une variable **affectée** possède un emplacement mémoire bien défini et connu de l'utilisateur par contre une variable **non affectée** ne possède pas une adresse figée, cette dernière peut changer à chaque cycle.

Une **variable publique** est une variable associée à certains blocs fonction et qui contient des valeurs qui ne sont pas influencées par le processus, elles peuvent par exemple être utilisées pour paramétrer le bloc fonction.

Une **variable privée** est une variable utilisée par le bloc fonction, lors du traitement de son algorithme, et qui est inaccessible à l'utilisateur.

Certains modules métiers dans Unity Pro possèdent ce que l'on appelle des I/ODDT (Input/Output Derived Data Type) qui sont des types de données structurées représentant soit une voie du module soit le module tout entier.

L'utilisateur peut aussi pendant la création de son projet utiliser des **constantes** qu'Unity Pro considérera comme des variables qui ne peuvent être modifiées pendant la phase d'exécution du programme.

3.3.2. Les types de donnés

Les données dans Unity Pro peuvent être soit des **variables** soit des **Blocs fonction**. Ces données sont organisées en familles comme le montre le schéma suivant :

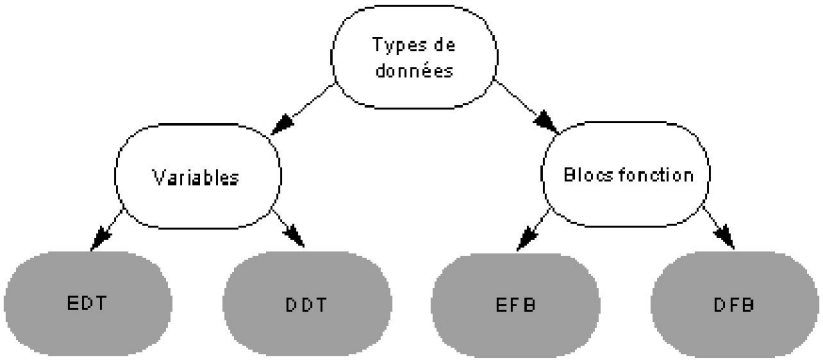


Fig. IV.16 familles de types de données

Ces différentes familles sont décrites par le tableau suivant :

Famille	Définition
Structures	Types de données élémentaires (Elementary data types) tels que: • Bool • Int • Byte • Mot • Dword • etc.
Tableaux	Types de données dérivés (Derived data types) tels que: • tableaux, qui contiennent des éléments de même type: ○ tableaux de Bool (tableaux d'EDT), ○ tableaux de tableaux (tableaux de DDT), ○ tableaux de structures (tableau de DDT), • structures, qui contiennent des éléments de type différents : ○ structures de Bool, Word, etc. (structures EDT) ○ structures de tableaux, structures de structures, structures de tableaux/structures (structures de DDT), ○ structures de Bool, structures de tableaux, etc. (structures EDT et DDT) ○ structures concernant les données d'entrées/de sorties (structures d'IODDT), ○ Structures contenant des variables restituant les propriétés et l'état d'une action ou transition d'un diagramme fonctionnel en séquence (Sequential Function Chart).
EFB	Blocs fonction élémentaires écrits en langage C. Ils comprennent : • des variables d'entrées, • des variables internes, • des variables de sorties, • un algorithme de traitement.
DFB	Blocs fonctions utilisateurs (Derived function blocs) écrits en langage d'automatisme (Litteral Structuré, Liste d'instructions, etc...). Ils comprennent : • des variables d'entrées, • des variables internes, • des variables de sorties, • un algorithme de traitement.

Tableau. II.1 Familles de variables dans Unity Pro

3.3.3. Adressage des variables dans Unity Pro

Une variable dans Unity Pro peut être adressée de différente manière selon qu'elle se trouve en mémoire interne de l'automate ou qu'elle est recueillie (ou émise) à travers un périphérique d'entrée (resp. de sortie).

La syntaxe d'une instance de données (en adressage direct) est définie par le symbole **%** suivi d'un préfixe de localisation mémoire et dans certains cas d'informations supplémentaires. Le préfixe de localisation mémoire peut être:

- M, pour les variables internes,
- K, pour les constantes (Premium et Modicon M340),
- S, pour les variables systèmes,
- N, pour les variables réseaux,
- I, pour les variables d'entrées,
- Q, pour les variables de sorties.

3.3.3.1 Variables internes %M

La syntaxe pour ce type de variables est définie par le tableau suivant :

	Syntaxe	Format	Exemple	Droit d'accès programme
Bit	%M<i> ou %MX<i>	3 bits (EBOOL)	%M1	R/W
Mot	%MW<i>	16 bits (INT)	%MW10	R/W
Bit extrait de Mot	%MW<i>.<j>	1 bit (BOOL)	%MW15.5	R/W
Double mot	%MD<i> (1)	32 bits (DINT)	%MD8	R/W
Réel (flottant)	%MF<i> (1)	32 bits (REAL)	%MF15	R/W
Légende				
(1): Non disponible pour Modicon M340.				

Tableau. II.2 variables internes

<i> représente le numéro de l'instance.

Pour le Modicon M340, les instances de type double (mot double) ou de flottement (réelles) doivent se situer dans un type d'entier %MW. L'index <i> de %MW doit être pair.

3.3.3.2. Les constantes %K

La syntaxe pour ce type de variables est définie par le tableau suivant :

	Syntaxe	Format	Droit d'accès programme
Constante Mot	%KW<i>	16 bits (INT)	R
Constante double mot	%KD<i> (1)	32 bits (DINT)	R
Constante Réel (flottant)	%KF<i> (1)	32 bits (REAL)	R
Légende			
(1): Non disponible pour Modicon M340.			

Tableau. II.3 les constantes %K

<i> représente le numéro de l'instance.

L'organisation de la mémoire est identique à celle des variables internes. Ces variables ne sont pas disponibles sur les automates Quantum

3.3.3.3. Les constantes %I

La syntaxe pour ce type de variables est définie par le tableau suivant :

	Syntaxe	Format	Droit d'accès programme
Constante Bit	%I<i>	3 bits (EBOOL)	R
Constante Mot	%IW<i>	16 bits (INT)	R

Tableau. II.4 les constantes %I

<i> représente le numéro de l'instance.

Ces données sont uniquement disponibles sur les automates Quantum et Momentum.

3.3.3.4. Les variables système %S

La syntaxe pour ce type de variables est définie par le tableau suivant :

	Syntaxe	Format	Droit d'accès programme
Bit	%S<i> ou %SX<i>	1 bit (BOOL)	R/W ou R
Mot	%SW<i>	32 bits (INT)	R/W ou R

Tableau. II.5 variables système

<i> représente le numéro de l'instance.

3.3.3.5. Les variables réseau %N

Ces variables contiennent des informations devant être échangées entre plusieurs programmes applicatifs à travers le réseau de communication.

La syntaxe pour ce type de variables est définie par le tableau suivant :

	Syntaxe	Format	Droit d'accès programme
Mot commun	%NW<n>.<s>.<d>	16 bits (INT)	R/W ou R
Bit extrait de Mot	%NW<n>.<s>.<d>.<j>	1 bit (BOOL)	R/W ou R

Tableau. II.6 variables réseau

<n> représente le numéro de réseau.

<s> représente le numéro de la station.

<d> représente le numéro de la donnée.

<j> représente le rang du bit dans le mot.

3.3.3.6. Les variables d'Entrée/Sortie %I et %Q

Ces variables sont contenus dans les modules d'entrée/sortie et les modules métier.

La syntaxe pour ce type de variables définie par le tableau suivant :

Entrées %I			
Bit d'erreur module de type BOOL	%I<@mod>.MOD.ERR	%I4.2.MOD.ERR	R
Bit d'erreur voie de type BOOL	%I<@mod>.<c>.ERR	%I4.2.3.ERR	R
Bit de type BOOL ou EBOOL	%I<@mod>.<c>	%I4.2.3	R
	%I<@mod>.<c>.<d>	%I4.2.3.1	R
Mot de type INT	%IW<@mod>.<c>	%IW4.2.3	R
	%IW<@mod>.<c>.<d>	%IW4.2.3.1	R
Double mot de type DINT	%ID<@mod>.<c>	%ID4.2.3	R
	%ID<@mod>.<c>.<d>	%ID4.2.3.2	R
Réel (flottant) de type REAL	%IF<@mod>.<c>	%IF4.2.3	R
	%IF<@mod>.<c>.<d>	%IF4.2.3.2	R
Sorties %Q			
Bit de type EBOOL	%Q<@mod>.<c>	%Q4.2.3	R/W
	%Q<@mod>.<c>.<d>	%Q4.2.3.1	R/W
Mot de type INT	%QW<@mod>.<c>	%QW4.2.3	R/W
	%QW<@mod>.<c>.<d>	%QW4.2.3.1	R/W
Double mot de type DINT	%QD<@mod>.<c>	%QD4.2.3	R/W
	%QD<@mod>.<c>.<d>	%QD4.2.3.2	R/W
Réel (flottant) de type REAL	%QF<@mod>.<c>	%QF4.2.3	R/W
	%QF<@mod>.<c>.<d>	%QF4.2.3.2	R/W

Tableau. II.7 variables d'Entrée/Sortie

<@mod = \<b>.<e>\<r>.<m>

<b> numéro de bus (omis si station locale).

<e> numéro du point de connexion de l'équipement (omis si station locale, le point de connexion est appelé aussi Drop pour les utilisateurs d'automates Quantum).

<r> numéro de rack.

<m> emplacement du module

<c> numéro de voie (0 à 999) ou mot réservé MOD.

<d> numéro donnée (0 à 999) ou mot réservé ERR (facultatif si valeur 0). Pour Modicon M340 <d> est toujours pair.

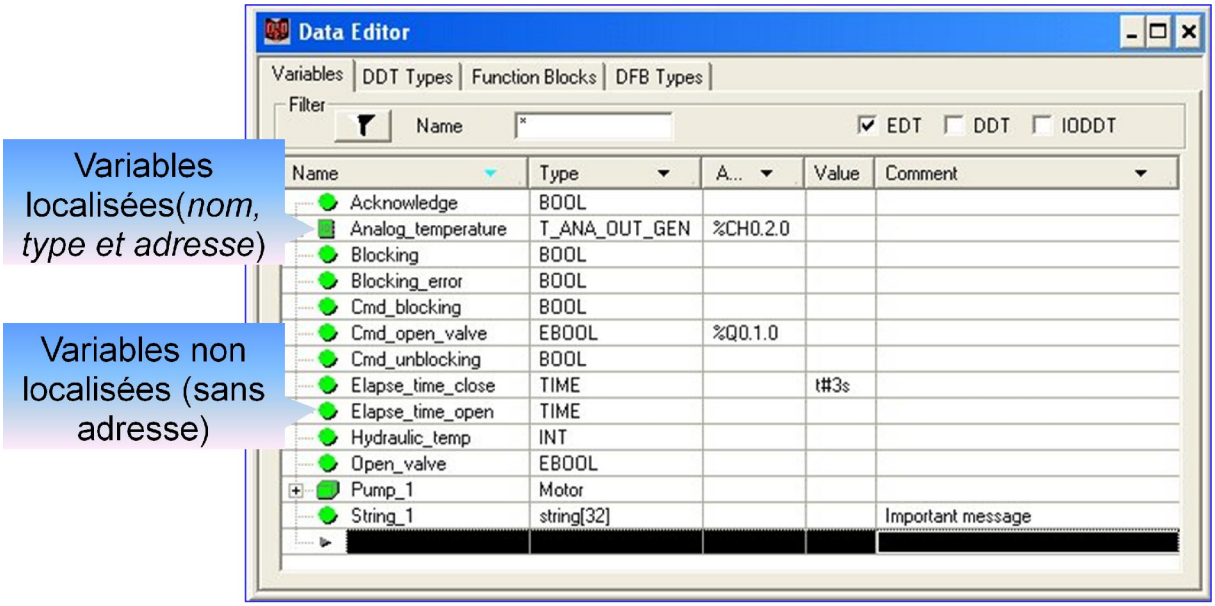
3.3.4. Edition des variables

Pour éditer les différentes variables utilisées dans le programme, il faut déployer le **répertoire** « Variables et instances FB » dans le navigateur de projet.

3.3.4.1. Onglet « variables élémentaires »

Cet onglet permet d'éditer aussi bien les variables localisées que non localisées.

On choisit pour ces variables leurs types (BOOL, INT, REAL…), un nom de variable, une adresse (pour les variables localisées) et un commentaire.



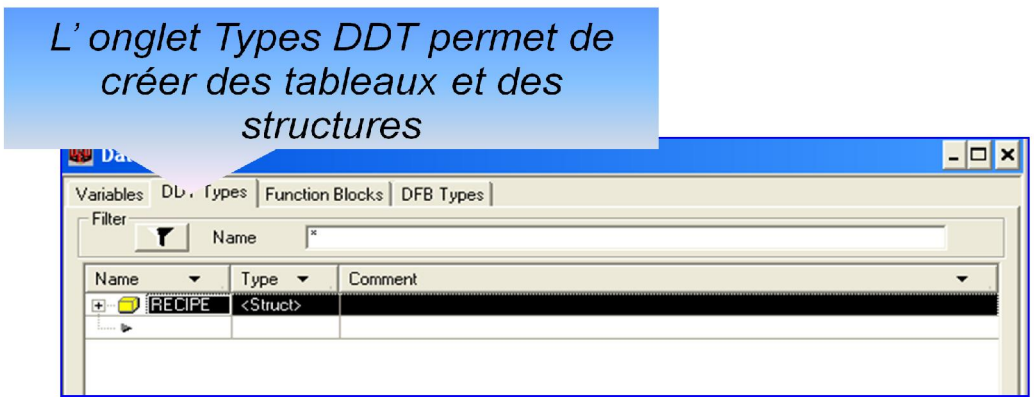


Fig. II.18 édition de variables dérivées

3.3.4.3. Onglet « variables dérivées E/S » ou IODDT

Ces variables peuvent représenter soit une voie d'un module d'E/S soit le module tout entier.

3.3.4.4. Onglet « Instances FB élémentaire » ou EFB

Cet onglet permet de définir les variables d'E/S associées à un bloc fonction élémentaire (EFB) se trouvant dans la bibliothèque (temporisation, compteurs…)

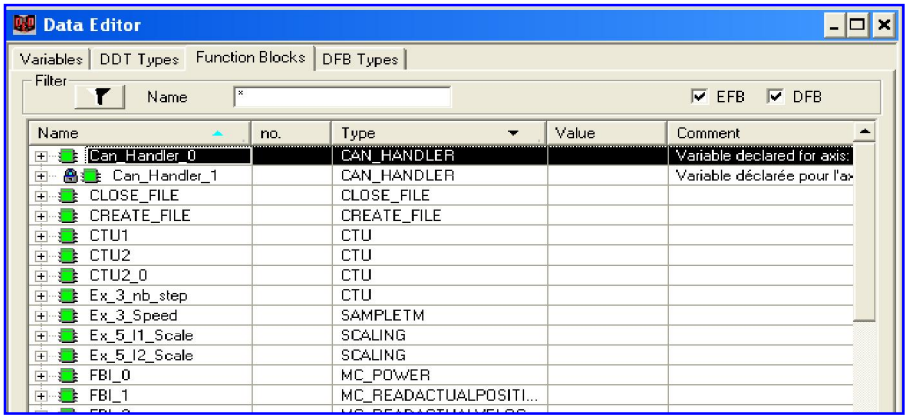


Fig. II.19 édition des instances du bloc fonction élémentaire (EFB)

3.3.4.5. Onglet « Instances FB dérivé » ou DFB

Cet onglet permet de créer un bloc fonction dérivé (DFB) et de définir les variables d'E/S locales ou globales associées. Ces DFB peuvent être crée en utilisant les cinq langages normalisés d'Unity Pro.

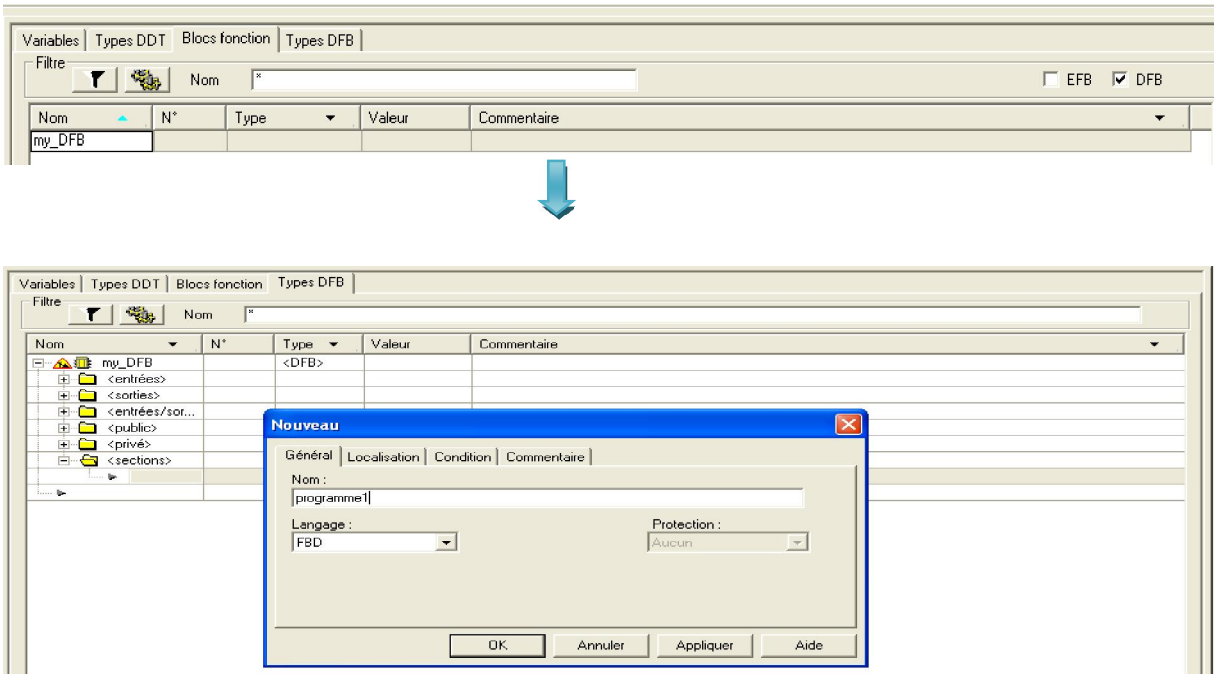


Fig. II.20 édition des instances du bloc fonction dérivé (DFB)

Le DFB ainsi crée ne peut être utilisé que dans le projet en cours. Pour pouvoir le réutiliser dans d'autres projets, il faudra l'ajouter à la bibliothèque comme l'explique la figure suivante :

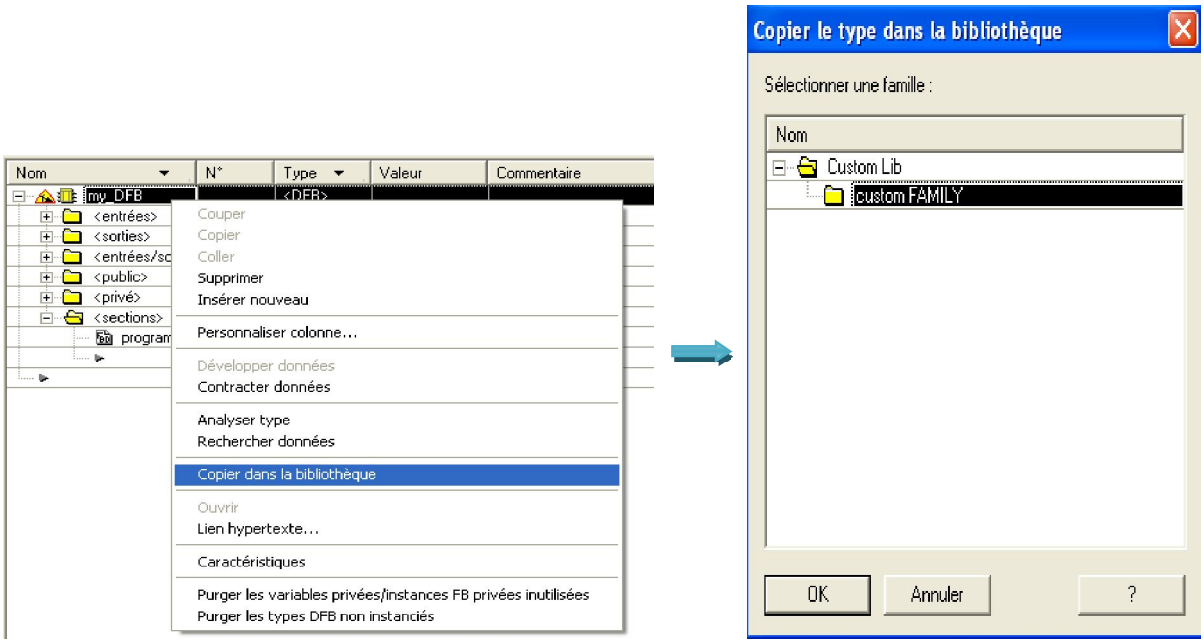


Fig. II.21 ajout du DFB à la Bibliothèque

3.4. Structuration du programme sous Unity Pro

On dit qu'Unity Pro est un logiciel **multitâche** c.à.d. que le programme utilisateur est réparti sur plusieurs parties (les tâches) et qui sont exécutées par ordre de priorité.

Il existe quatre types de tâches sous Unity Pro qui sont : tâches MAST (tâche principale), tâches rapides (FAST), tâches événementielles (EVT) et les tâches auxiliaires (AUX).

L'ordre de priorité de ces tâches est illustré par la **figure II.22**

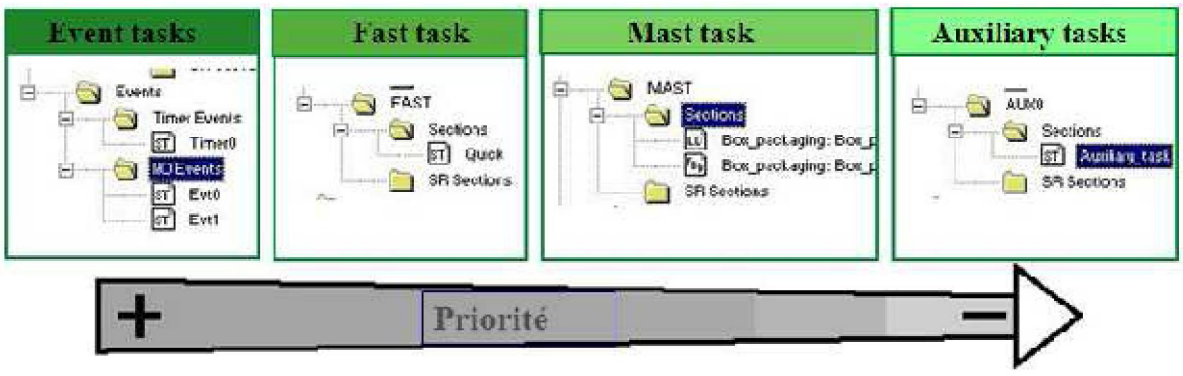


Fig. II.22 ordre de priorité des différentes tâches

3.4.1. Tâche événementielle (EVT)

Certains événements du process à superviser doivent recevoir l'attention de l'automate dés leur apparition (ou du moins dans un délai très court). Dans ce cas, on utilise les tâches événementielles pour réduire le temps de réponse du programme aux événements des Modules d'entrée/sortie et aux temporisateurs d'événement.

3.4.2. Tâche rapides (FAST)

C'est une tâche périodique de période configurable de 1 à 255 ms. Sa programmation est analogue à celle de la tâche principale et le contrôle ainsi que la visualisation des temps d'exécution se fait au travers des bits système.

3.4.3. Tâche principale (MAST)

MAST est l'abréviation de Master (maitre) ce qui sous entend que le programme principal est écrit dans cette tâche. Le programme de la tâche maître est constitué de plusieurs modules de programmes appelés sections et sous-programmes (appelés par les sections ou depuis un autre sous programme), qui sont liées à une tâche.
On peut configurer l'exécution de la tâche MAST de façon à ce que le programme s'exécute de manière cyclique ou périodique.

3.4.4. Tâche auxiliaire (AUX)

Ces tâches sont utilisées pour traiter des événements plus lents que ceux des tâches précédentes et donc qui requièrent le processeur beaucoup plus longtemps.

Ces tâches sont périodiques et la période est configurable entre 10 ms à 2,55 s.

La tâche qui vient par défaut sous Unity Pro est la tâche MAST. Pour ajouter une nouvelle tâche, on va vers le navigateur de projet et on suit les étapes illustrées par la **figure IV.23.**

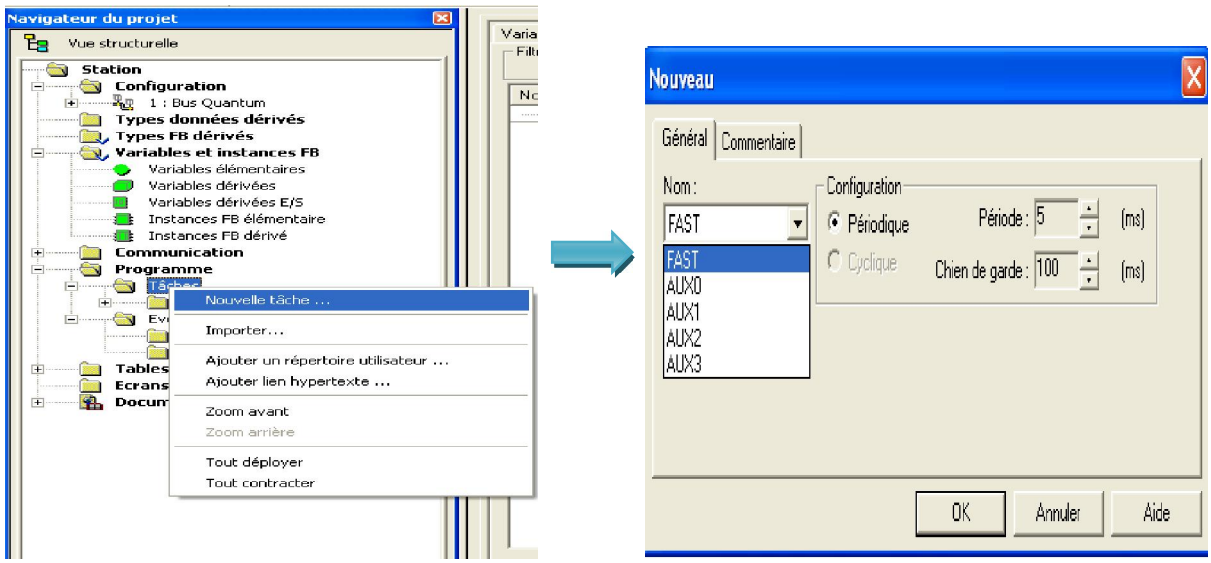


Fig. II.23 Création d'une nouvelle tâche

3.5. Edition du programme utilisateur

Comme on l'a dit précédemment, le programme utilisateur est stocké dans des sections et des sous programmes (appelés par les sections ou depuis un autre sous programme).

Pour ajouter une nouvelle section à la tâche MAST, on déploie le répertoire de cette dernière dans le navigateur de projet et on clique avec le bouton droit de la souris sur « sections » puis on choisit « nouvelle section », après quoi une nouvelle fenêtre apparait dans l'éditeur de programme pour choisir le nom de la section et le langage de programmation de cette dernière.

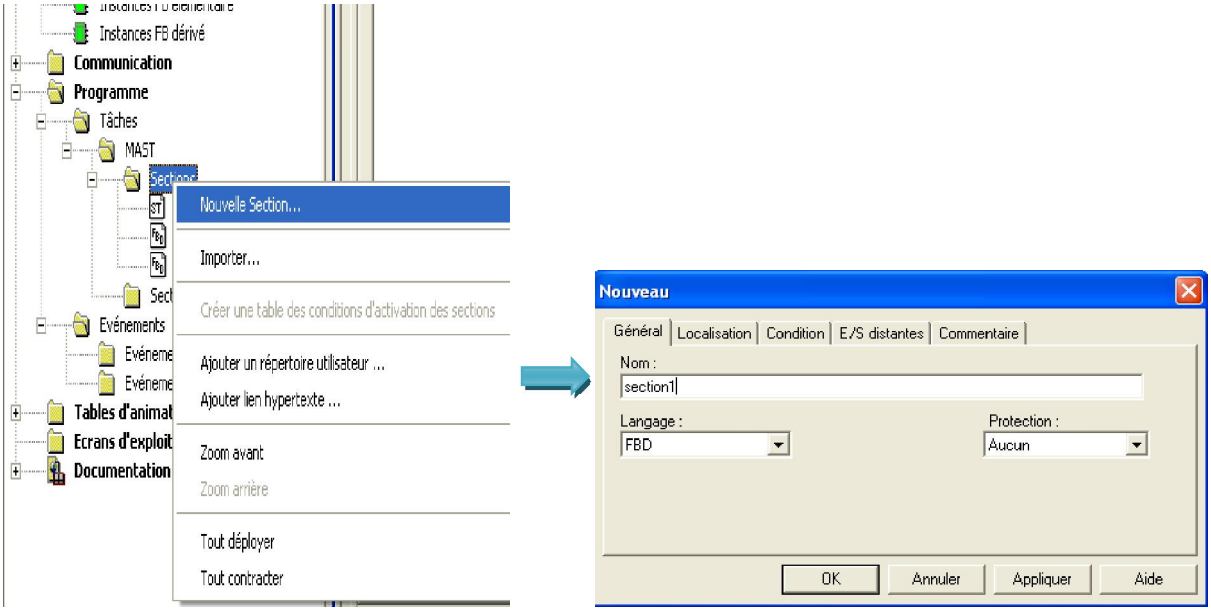


Fig. II.24 Création d'une nouvelle section

3.5.1. Edition du programme en FBD (Functional Block Diagram)

Le langage FBD est un langage graphique conforme à la norme IEC 61131-3 et qui est très bien adapté à la commande des processus. Comme son nom l'indique, un programme écrit en FBD est composé de blocs fonctionnels élémentaires reliés les uns aux autres par des lignes sur lesquelles circule le flux de données. Un exemple de programme écrit en FBD est donné à la **figure IV.25**.

Le programme FBD est écrit dans une section formée de 30 colonnes et 23 lignes

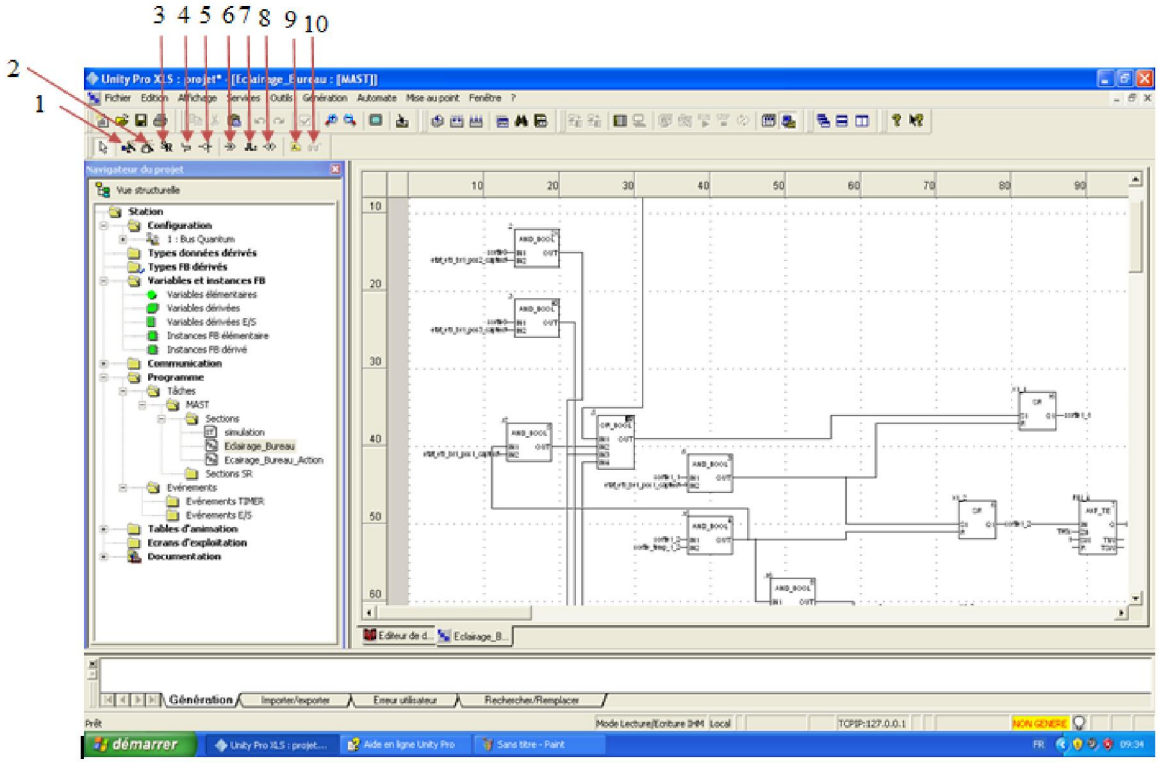


Fig. II.25 Exemple d'un programme écrit en FBD

1) Sélection de données : Pour choisir une variable déjà créée dans la bibliothèque ou bien définir une nouvelle variable.

2) Assistant saisie FFB : Pour entrer un nouveau bloc issu de la bibliothèque.

3) Sous programme : Pour faire un appel à un sous programme.

4) Liaison : Pour relier les différents blocs

5) Négation de branche

6) Saut vers une autre partie du programme

7) Etiquette de saut

8) Retour vers le programme principal

9) Commentaire.

10) fenêtre d'inspection.

Pour importer des valeurs dans le FFB ou exporter des valeurs du FFB, des entrées et des sorties sont nécessaires. Elles sont appelées paramètres formels.

Les paramètres formels sont liés à des objets qui comprennent les états courants du traitement. Ces objets sont appelés paramètres réels.

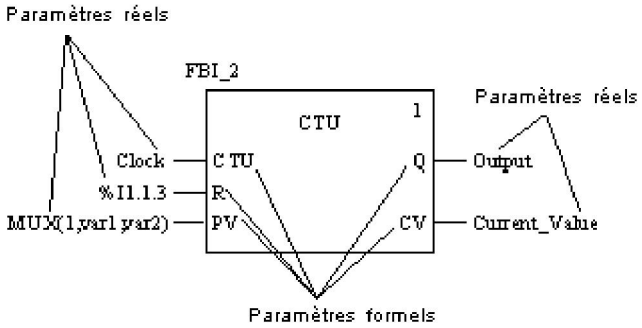


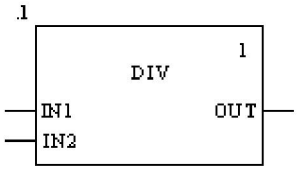
Fig. II.26 Paramètres d'un bloc FBD

D'une manière générale le langage FBD utilise des FFB qui est le terme commun pour désigner

- Les fonctions élémentaires (EF)
- Les blocs fonction élémentaires (EFB)
- Les blocs fonction dérivés (DFB)
- Les procédures.

3.5.1.1. Les fonctions élémentaires (EF)

Les fonctions élémentaires (EF) n'ont pas d'état interne. Pour des valeurs égales aux entrées, la valeur à la sortie est la même à chaque exécution de la fonction. Par exemple, l'addition de deux valeurs donne toujours le même résultat.



Une fonction élémentaire est représentée graphiquement sous forme de cadre avec des entrées et une sortie. Les entrées sont toujours représentées sur la gauche et la sortie toujours sur la droite du cadre.

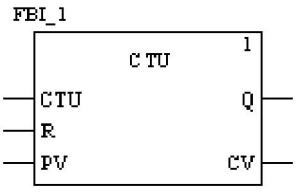
Le nom de la fonction, c'est-à-dire le type de fonction, est affiché au centre du cadre.

Le numéro d'exécution de la fonction apparaît à droite du type de fonction.

Le numéro de fonction est affiché au-dessus du cadre. Le numéro de fonction représente le numéro courant de la fonction dans la section actuelle. Les numéros de fonction ne peuvent pas être modifiés.

3.5.1.2. Les blocs fonction élémentaires (EFB)

Les blocs fonction élémentaires (EFB) ont des états internes. Pour des valeurs égales aux entrées, la valeur à la sortie peut être différente pour toutes les exécutions de la fonction.



Un bloc fonction élémentaire est représenté graphiquement sous forme de cadre avec des entrées et des sorties et il a les même caractéristiques (graphiques) que les EF.

Le nom d'instance qui permet d'identifier précisément le bloc fonction dans un projet est affiché au-dessus du cadre.

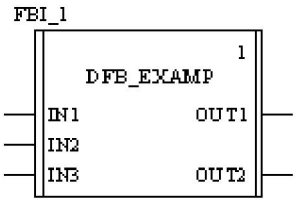
Ce nom d'instance est généré automatiquement et présente la structure suivante : FBI_n

FBI = Instance de bloc fonction

n = numéro courant du bloc fonction au sein du projet

3.5.1.3. Les blocs fonction dérivés (DFB)

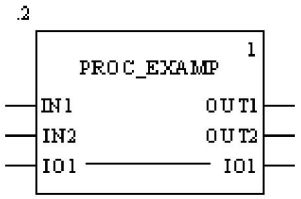
Les blocs fonction dérivés (DFB) ont les mêmes caractéristiques que les blocs fonction élémentaires. Ils sont cependant créés par l'utilisateur dans les langages FBD, LD, IL et/ou ST.



L'unique différence par rapport aux blocs fonction élémentaires est que le bloc fonction dérivé est représenté graphiquement sous forme de cadre avec deux lignes verticales.

3.5.1.4. Les procédures

Techniquement, les procédures sont des fonctions.



L'unique différence par rapport aux fonctions élémentaires est que les procédures peuvent comprendre plus d'une sortie et qu'elles prennent en charge le type de données VAR_IN_OUT.

3.5.2. Edition du programme en LD (Ladder Diagram)

Le langage Ladder est un langage à contact qui traduit directement les équations booléennes en circuit électrique dont les contacts (N.O ou N.F) sont les entrées et les relais sont les sorties. Ce circuit est délimité à gauche par une barre représentant l'alimentation (AC ou DC) et à droite par le neutre qui est non visible mais connecté logiquement aux FFB et sorties des bibines. Un exemple de programme écrit en Ladder est donné par la **figure IV.27.**

Le langage de programmation LD est orienté cellules, c'est-à-dire qu'un seul objet est placé dans chaque cellule. Une grille divise la fenêtre en lignes et colonnes. L'ordre de traitement des objets individuels d'une section LD est déterminé par le flux de données d'une section. Les réseaux associés à la barre d'alimentation gauche sont traités du haut vers le bas.

L'éditeur de programme LD est illustré à la **figure II.28**.

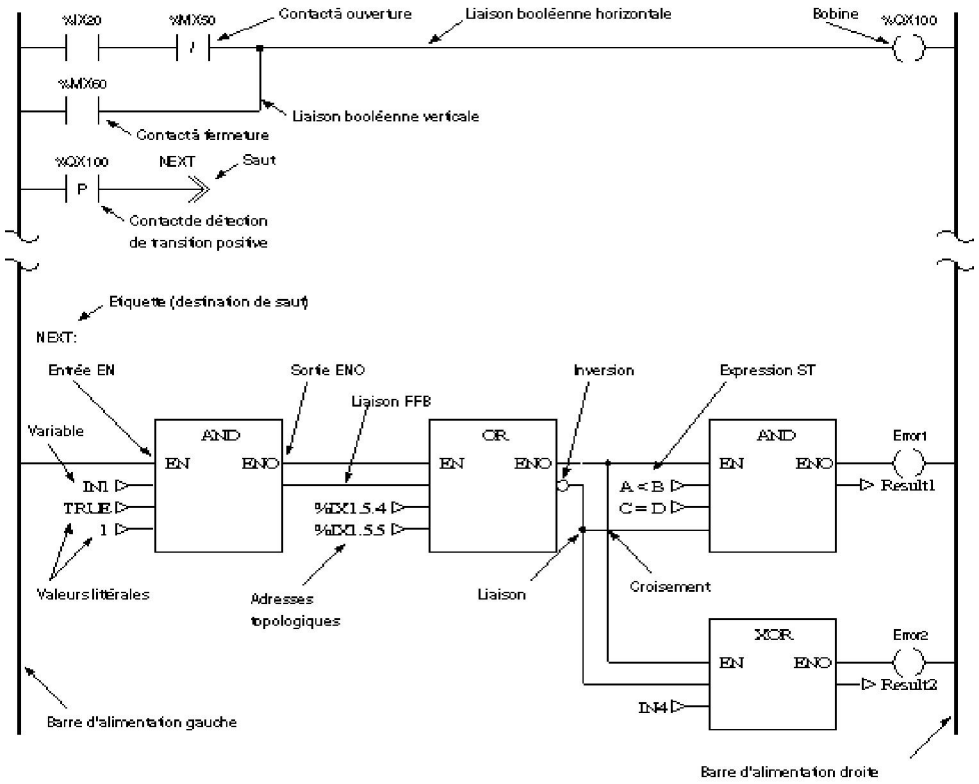


Fig. II.27 exemple d'un programme écrit en Ladder

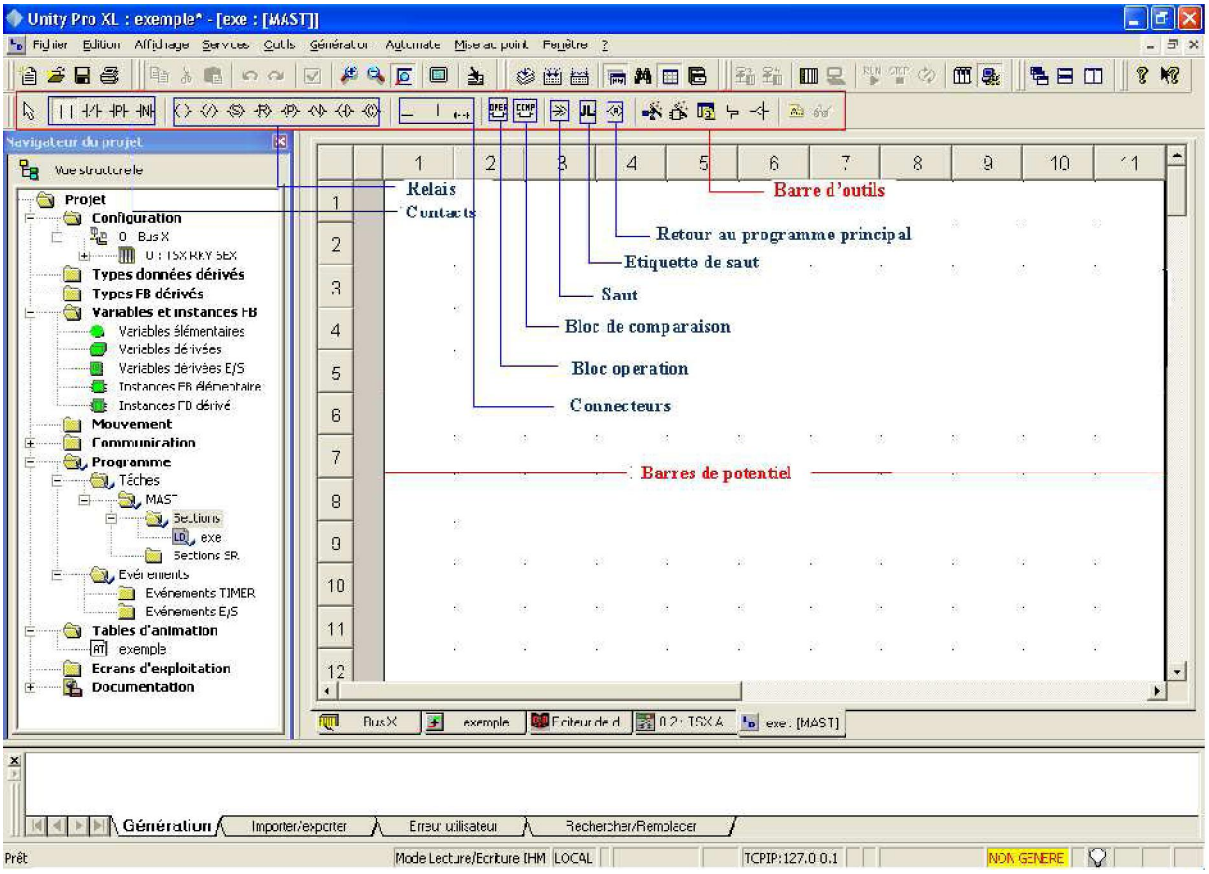


Fig. IV.28 éditeur de programme LD

3.5.3. Edition du programme en SFC (Sequencial Flow Chart)

Le langage SFC est équivalent au langage GRAFCET. Un programme écrit en SFC se compose de sections SFC (niveau supérieur), de sections de transition et de sections d'actions. Dans Unity Pro, les programmes SFC ne sont supportés que par la tâche MAST. Pour les autres tâches (AUX, EVT et FAST) on utilisera d'autres langages de programmation.
Une section SFC contient les objets de création de programme suivants :

- Étape
- macro étape (séquence de sous-étapes imbriquées)
- transition (condition de transition)
- saut
- liaison
- divergence en OU
- convergence en OU
- divergence en ET
- convergence en ET

Ces objets sont représentés à la **figure II.29**

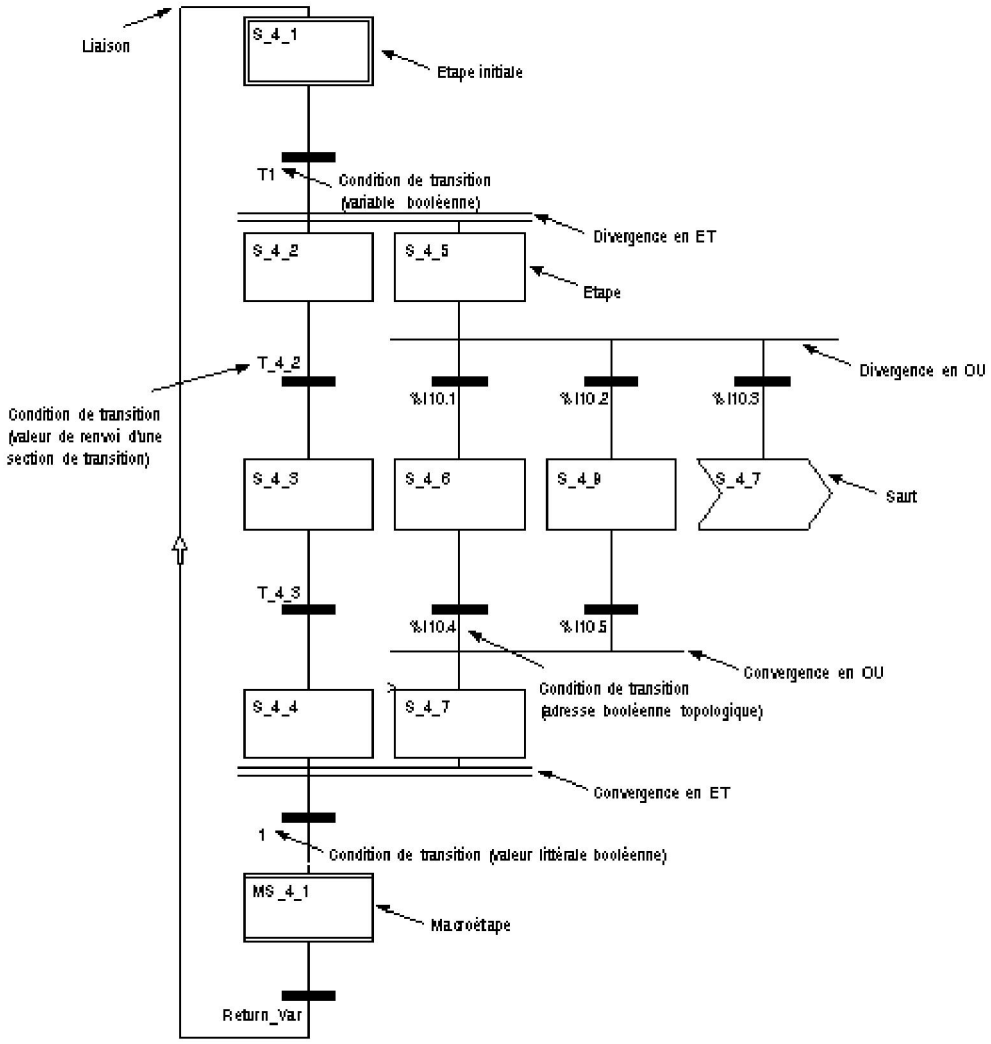


Fig. II.29 exemple d'un programme écrit en SFC

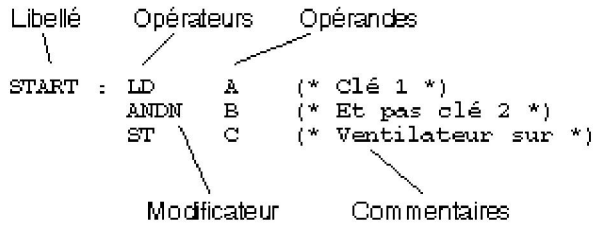
Dans un programme écrit en SFC, deux étapes consécutives ne peuvent jamais être directement reliées ; elles sont toujours séparées par une transition. Le passage d'un état à un autre est déclenché par l'activation d'une transition. Le déroulement d'une séquence va dans le sens des liaisons dirigées et se déroule de la partie inférieure de l'étape précédente à la partie supérieure de l'étape suivante (de la même façon que le GRAFCET). Les divergences sont traitées de gauche à droite.
Chaque étape peut compter zéro ou plusieurs actions. A chaque transition est associée une condition de transition.
La dernière transition de la séquence est toujours reliée à une autre étape de la séquence (par une liaison graphique ou un symbole de "saut") de manière à obtenir une boucle fermée.

3.5.4. Edition du programme en LIST (IL)

Le langage IL est un langage textuel de bas niveau (proche du langage machine), qui utilise un jeu d'instructions simples. Il trouve sa puissance dans les applications de petites tailles, et dans la création de sous programme ou procédure, car il permet un contrôle totale et une optimisation parfaite du code ; par contre pour les grandes applications il est très difficile de programmer avec le IL ; les programmes dans ce langage peuvent être traduit ou déduit des autres langages.

Le langage LIST a la même structure que l'assembleur ; il utilise un ou plusieurs registres de travail. Les valeurs intermédiaires nécessaires pour l'exécution d'une instruction donnée seront mémorisées dans ces registres le temps de leur utilisation et il possède un jeu d'instructions assez riche pour décrire toutes les opérations arithmétiques et logiques, les opérations de comptage et temporisation, la comparaison et le transfert…

Exemple :



Dans Unity Pro, on ne peut avoir qu'une seule instruction par ligne. Chaque instruction a la structure suivante :

Instruction		
Partie type opération	Partie opérande	
	Identificateur	Paramètre
AND	%M	1

Tableau. II.8 Structure d'une instruction en LIST

- **Partie type opération :** définit la nature de l'instruction. Les différents opérateurs sont :
  - § Assignment (*LD, ST, S, R*).
  - § Logique (*AND, OR, XOR, NOT*).
  - § Arithmétique (*ADD, SUB, MUL, DIV, MOD*).
  - § Comparaison (*GT, GE, EQ, NE, LE, LT*).
  - § Structure (*JMP, RET, «) »*).
  - § Appel de fonction (*CAL function_name*).

- **Partie opérande :** Contient les informations nécessaires à l'exécution de l'instruction et ceci en spécifiant l'objet sur lequel doit porter l'instruction.

3.5.5. Edition du programme en Littéral Structure (ST)

Le langage ST (Structured Text) est un langage de programmation textuel de haut niveau dédié aux applications d'automatisation ; il est utilisé principalement pour décrire les procédures complexes et difficilement modélisables avec les langages graphiques. Il peut aussi être utilisé en tant que sous programme avec d'autres langages de programmation.

Il utilise les mêmes énoncés que les langages de programmation de haut niveau (Pascal, C,C++…) comme: les assignations, les appels de fonction, les énoncés de contrôle (IF, THEN, ELSE, CASE) ou d'itération (FOR, WHILE, REPEAT), en plus des opérations arithmétiques et logiques.

On a accès directement aux boucles conditionnelles à travers la barre d'outils, on clique sur l'une d'elles et elle est automatiquement ajoutée dans notre programme. Il nous suffit de remplir les vides avec des instructions se terminant par un « ; ».

L'éditeur de programme ST est représenté à la figure suivante :

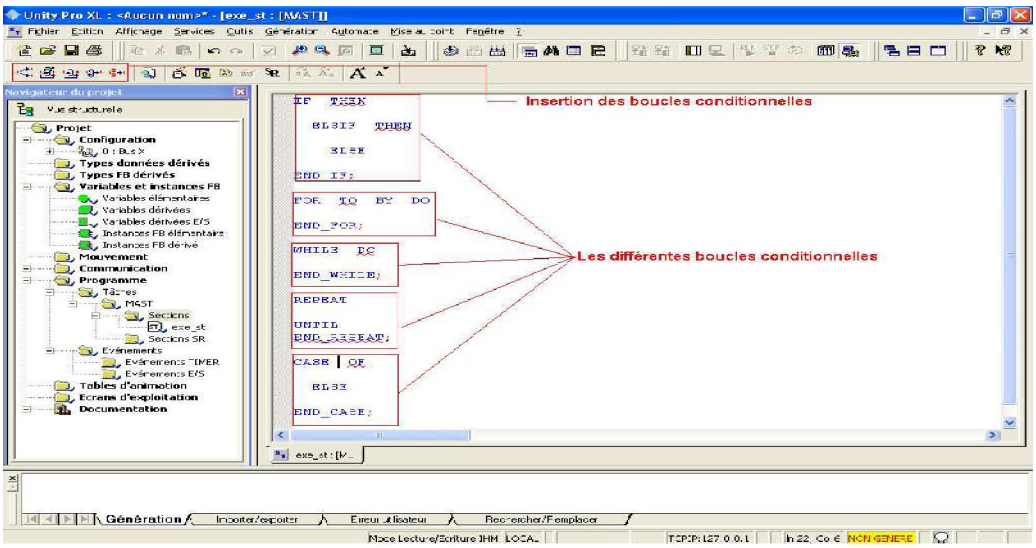


Fig. II.30 éditeur de programme ST

3.6. Compilation, simulation et transfert du programme vers l'automate

La première chose à faire de simuler le programme utilisateur et son transfert vers l'automate est de générer (compiler) le programme et ceci afin de vérifier qu'il n'y a pas d'erreurs aussi bien dans la configuration matérielle que dans la syntaxe du programme.

La génération du programme est illustrée à la figure suivante :

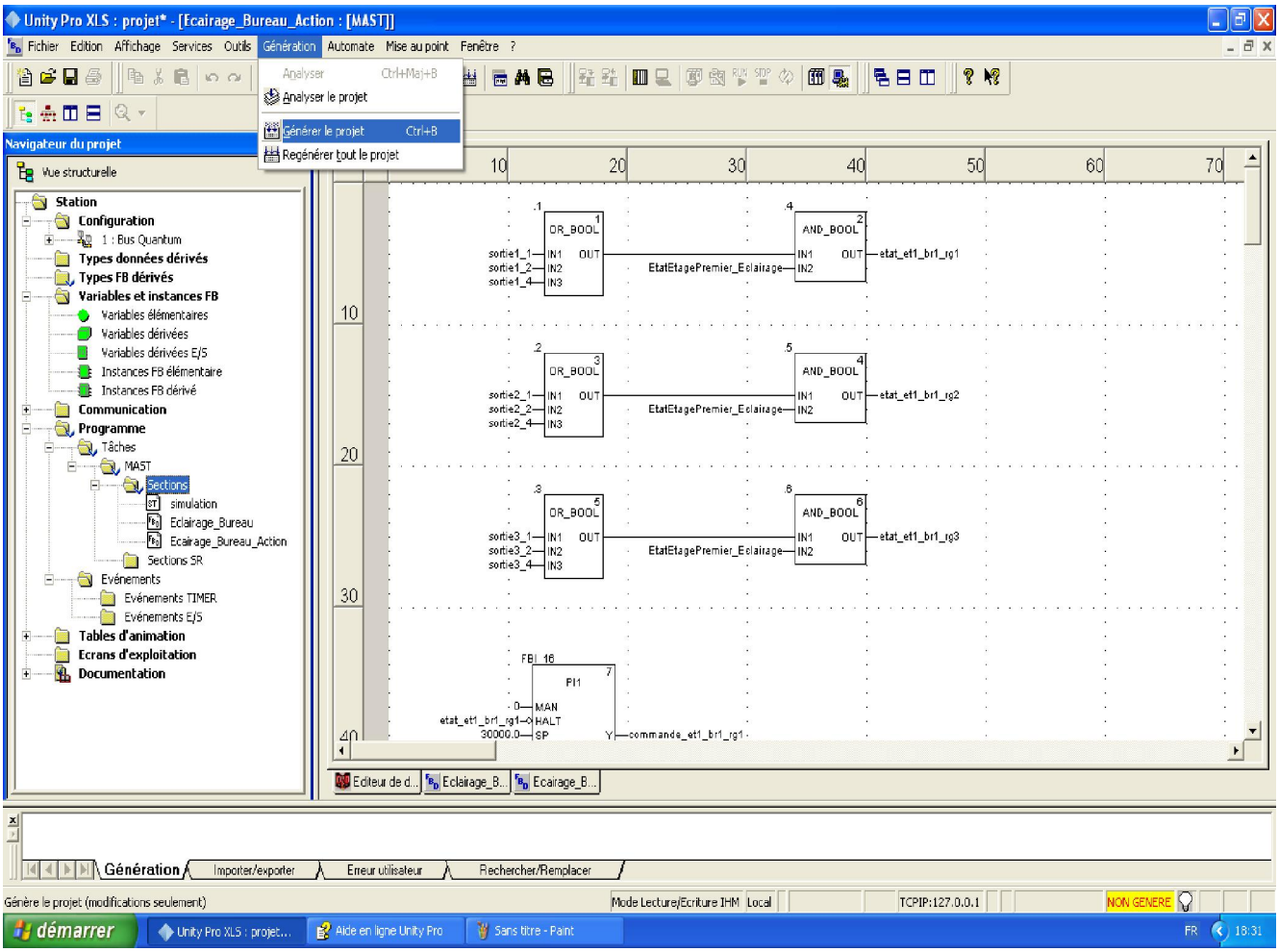


Fig. II.31 génération du projet

Une fois généré et si le projet ne comporte pas d'erreurs, la partie inférieure d'Unity Pro apparait comme ceci :

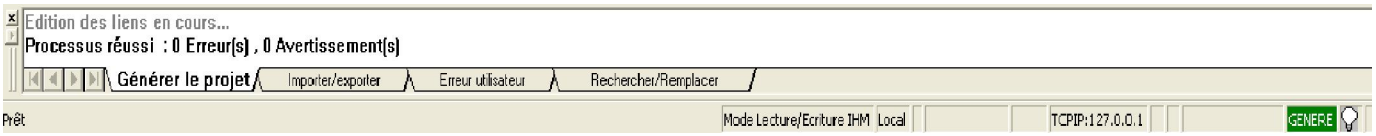


Fig. II.32 génération sans erreurs

Dès lors, on peut commencer la simulation du projet. Pour se faire, on connecte d'abord l'automate comme le montre la figure suivante :

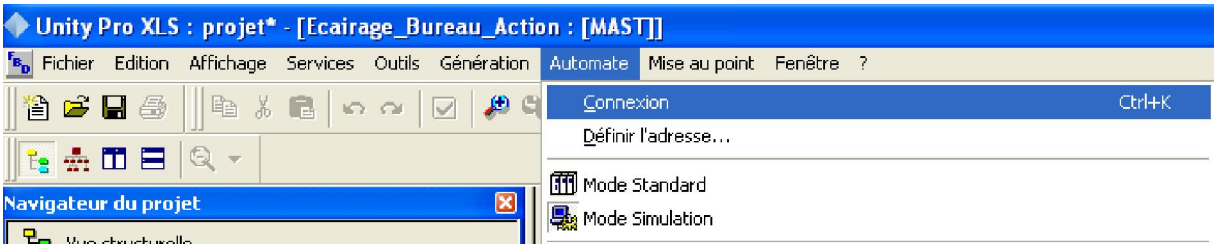


Fig. II.33 connexion de l'automate

Puis, on transfert le projet vers l'automate comme le montre la **figure IV.34**.

Il est à noter que l'adresse IP de l'automate (en mode local c.à.d.) lors de la simulation est 127.0.0.1.

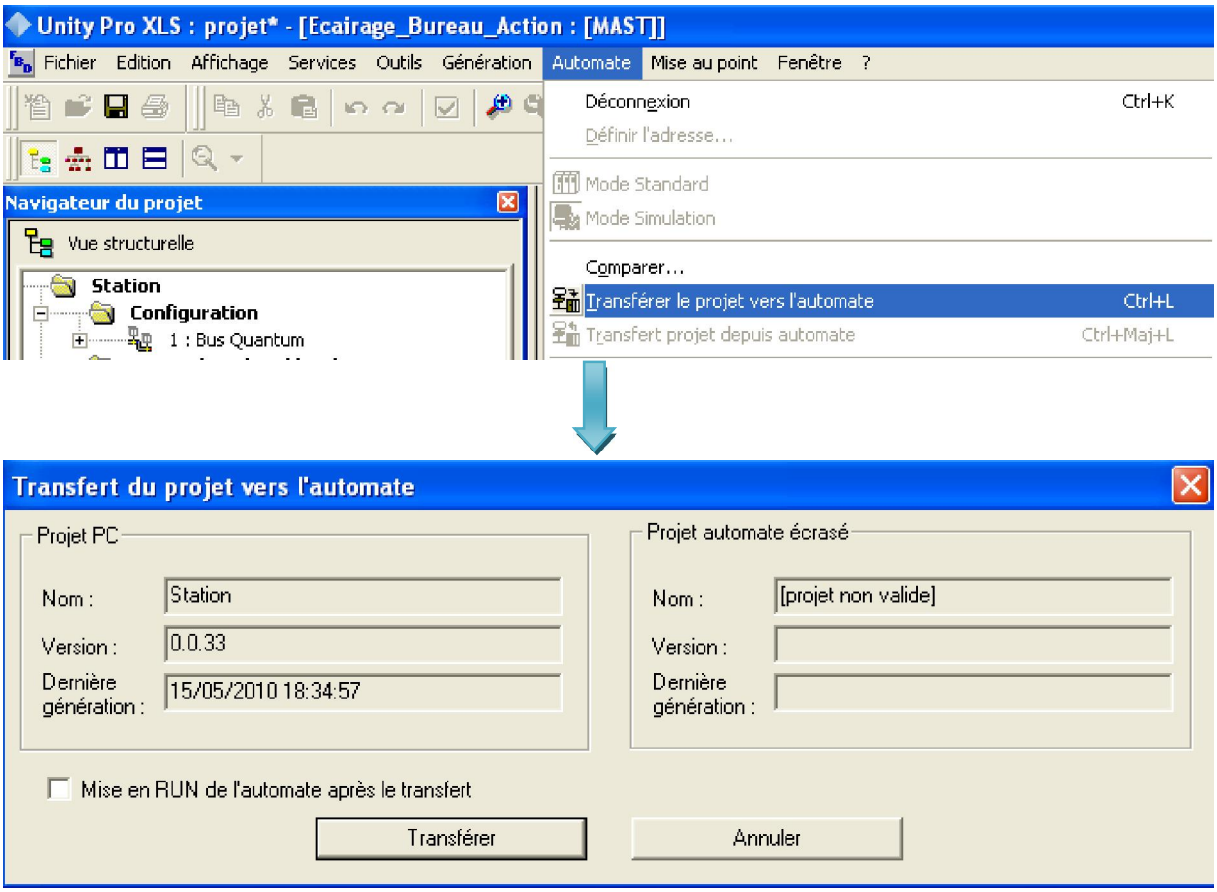


Fig. II.34 transfert du projet vers l'automate

On peut visualiser les l'état des variables de l'automate en créant une table d'animation comme le montre la figure suivante :

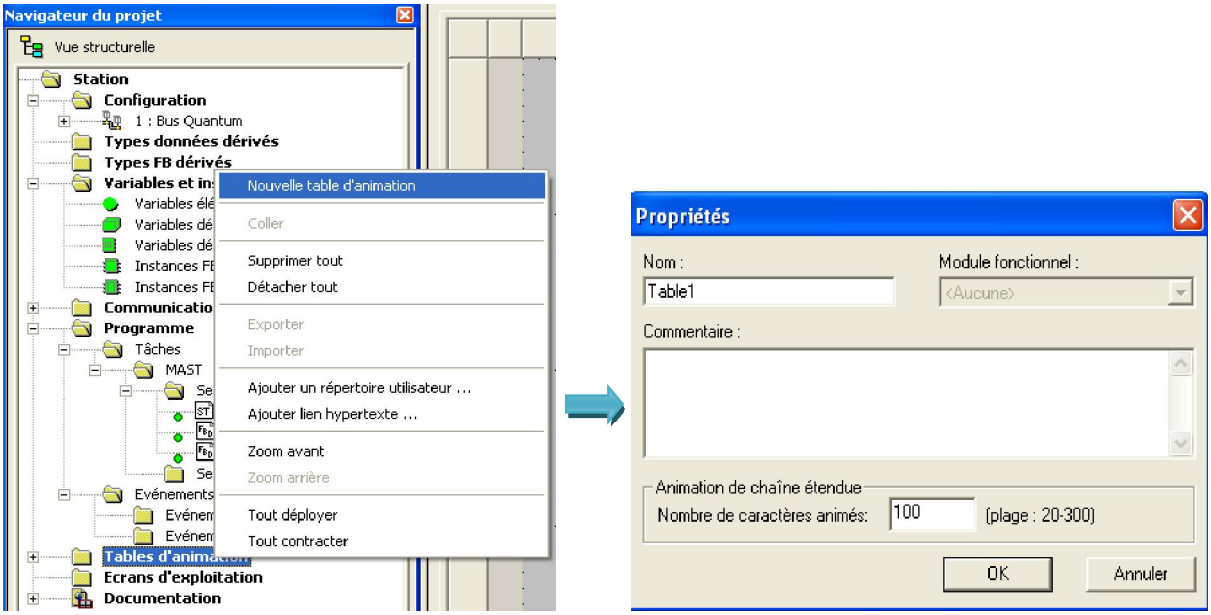
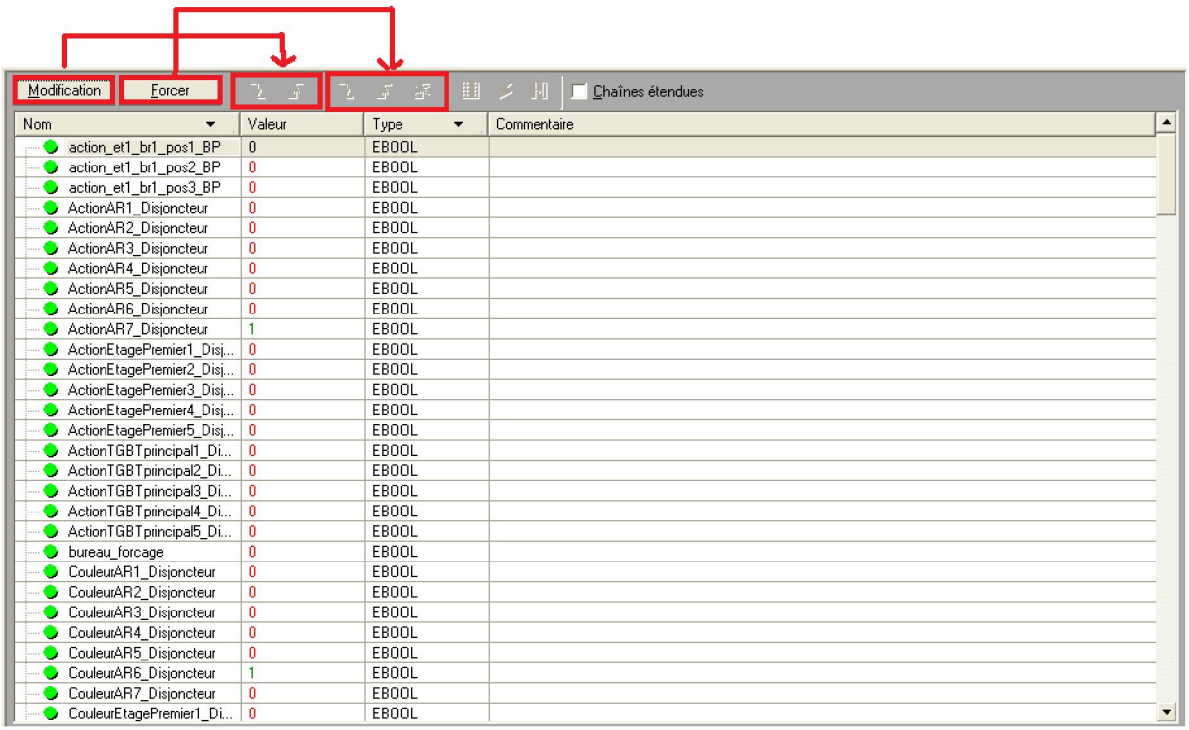


Fig. II.35 création d'une nouvelle table d'animation

Une fois crée, cette table doit être remplie avec les variables du projet. On peut ainsi visualiser les états des ces variables et les forcer à 1 ou à 0 (pour les variables booléennes) et à des valeurs entières ou réelles pour les autres variables.

Un exemple de table d'animation est illustré par la figure suivante :



Nom	Valeur	Type	Commentaire
action_et1_br1_pos1_BP	0	EBOOL	
action_et1_br1_pos2_BP	0	EBOOL	
action_et1_br1_pos3_BP	0	EBOOL	
ActionAR1_Disjoncteur	0	EBOOL	
ActionAR2_Disjoncteur	0	EBOOL	
ActionAR3_Disjoncteur	0	EBOOL	
ActionAR4_Disjoncteur	0	EBOOL	
ActionAR5_Disjoncteur	0	EBOOL	
ActionAR6_Disjoncteur	0	EBOOL	
ActionAR7_Disjoncteur	1	EBOOL	
ActionEtagePremier1_Disj...	0	EBOOL	
ActionEtagePremier2_Disj...	0	EBOOL	
ActionEtagePremier3_Disj...	0	EBOOL	
ActionEtagePremier4_Disj...	0	EBOOL	
ActionEtagePremier5_Disj...	0	EBOOL	
ActionTGBTprincipal1_Di...	0	EBOOL	
ActionTGBTprincipal2_Di...	0	EBOOL	
ActionTGBTprincipal3_Di...	0	EBOOL	
ActionTGBTprincipal4_Di...	0	EBOOL	
ActionTGBTprincipal5_Di...	0	EBOOL	
bureau_forcage	0	EBOOL	
CouleurAR1_Disjoncteur	0	EBOOL	
CouleurAR2_Disjoncteur	0	EBOOL	
CouleurAR3_Disjoncteur	0	EBOOL	
CouleurAR4_Disjoncteur	0	EBOOL	
CouleurAR5_Disjoncteur	0	EBOOL	
CouleurAR6_Disjoncteur	1	EBOOL	
CouleurAR7_Disjoncteur	0	EBOOL	
CouleurEtagePremier1_Di...	0	EBOOL	

Fig. II.36 Exemple d'une table d'animation

- **Modification :** Une variable peut être modifiée manuellement par l'utilisateur mais le programme peut aussitôt la remodifier à son tour.

- **Forcer :** La variable peut être modifiée manuellement par l'utilisateur et le programme n'a plus aucune influence sur cette variable.

4. Exemple d'application

4.1. Cahier de charge

Nous disposons d'un bureau muni de trois rangées de luminaires pour l'éclairage et de trois détecteurs de mouvement (chacun associé à une rangée) servant comme détecteur de présence. On veut que l'éclairage soit commandé par le détecteur de mouvement afin de l'éteindre lorsqu'aucune présence n'est enregistrée pour rendre ainsi la consommation d'énergie plus rationnelle.

- Une rangée de luminaire est allumée si le détecteur de mouvement associé détecte une présence, sinon elle est éteinte au bout de 2 mn après la dernière détection de mouvement.
- De plus, le bureau dispose aussi de trois boutons poussoirs permettant l'allumage d'une rangée de luminaire même si le capteur de mouvement associé est désactivé (aucune présence n'est détectée), ces boutons peuvent être utiles lorsqu'un occupant souhaite allumer d'autres rangées que celle sous laquelle il travaille dans le but d'augmenter la luminosité.
- L'allumage d'une rangée par un bouton poussoir sera activé pour une durée déterminée (20 mn), au bout de cette durée, les luminaires de la rangée vont s'éteindre dans le cas où le bureau est vacant.
- La lumière des luminaires les plus proches de la fenêtre est modulée (régulée) en fonction de la lumière du jour et ceci en utilisant un capteur de lumière et un régulateur PI.

Le GRAFCET matérialisant ce cahier de charge est illustré à la **figure II.38**

Le bureau tel qu'il a été conçu sous vijeo citect (que nous verrons plus loin) est illustré à la **figure II.37**

- **Capt :** désigne le capteur de mouvement pour chacune des rangées (1,2 et 3)
- **BP :** indique un bouton poussoir.
- **rg1 et rg2 :** si l'une de ces sorties vaut 1 alors il y aura allumage de la rangée de luminaire correspondante.
- **reg3 :** cette sortie si elle est à 1, activera la régulation de la lumière de la 3^{eme} rangée de luminaires en fonction de la lumière du jour.
- **LT1 à LT6 :** lancement d'une temporisation.
- **T1 à T6 :** fin de la temporisation.

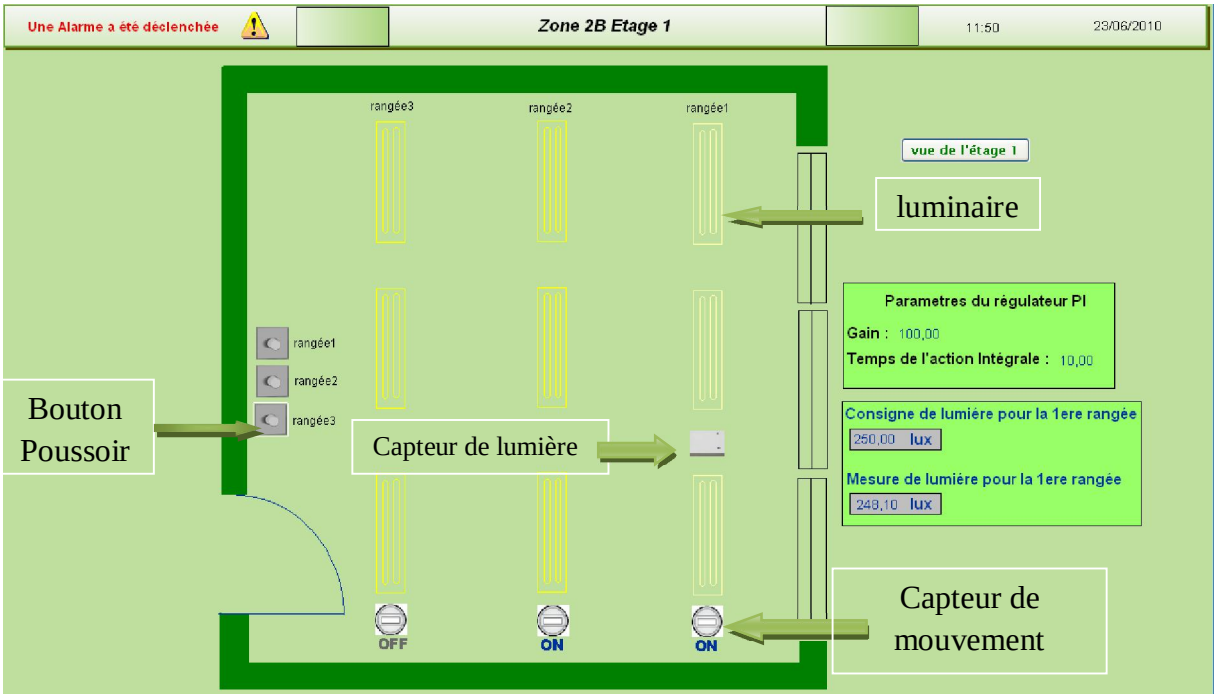


Fig. II.37. Représentation du bureau

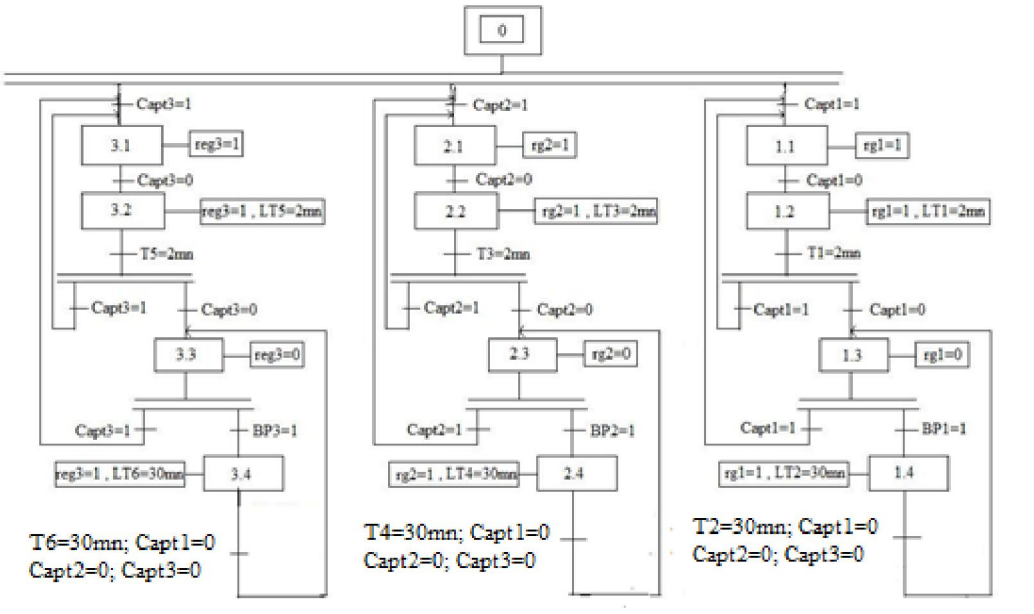


Fig. II.38 GRAFCET de l'exemple du bureau

4.2. Traduction du GRAFCET en équations logiques

Nous allons programmer le GRAFCET précédent en langage FBD. Pour cela, nous allons affecter à chaque état une bascule RS ayant comme entrée S (Set) et R (Reset) dont nous allons donner les équations logiques.

Puisque le GRAFCET précédent est constitué de 3 branches semblables, nous allons uniquement donner les équations de la 1ere branche :

$$X_0 \begin{cases} S_0=0 \\ R_0=(Capt1+Capt2+Capt3).X_0 \end{cases}$$

$$X_{1.1} \begin{cases} S_{1.1}=(Capt1).X_0+(Capt1)R_{1.2}+(Capt1).X_{1.3} \\ R_{1.1}=(\overline{Capt1})\ X_{1.1} \end{cases}$$

$$X_{1.2} \begin{cases} S_{1.2}=R_{1.1} \\ R_{1.2}= (T1=2mn).X_{1.2} \end{cases}$$

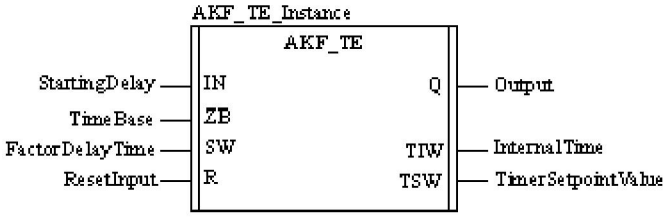
$$X_{1.3} \begin{cases} S_{1.3}=(\overline{Capt1})\ .R_{1.2} \\ R_{1.3}= (BP1+Capt1).X_{1.3} \end{cases}$$

$$X_{1.4} \begin{cases} S_{1.4}=(BP1).X_{1.3} \\ R_{1.4}=(T2=30mn).X_{1.4} \end{cases}$$

4.3. Programmation de l'exemple en FBD

Pour pouvoir programmer notre exemple en FBD, nous aurons besoin de définir deux principaux éléments à utiliser : la temporisation et le régulateur PI.

4.3.1. La temporisation



La temporisation que nous avons utilisé pour ce programme se nomme dans Unity Pro **AKF_TE** qui est une temporisation à l'enclenchement.

Le temps de retard est constitué d'une base temps **ZB** et d'un facteur **SW**. Le temps de retard est le résultat du produit **ZB x SW**.

La sortie **TIW** indique la valeur actuelle (mesure), qui se calcule de la manière suivante : **TIW= temps écoulé / ZB**

Un front montant sur l'entrée **IN** valide le retard sur la sortie **TSW**, le temporisateur est lancé et l'état actuel apparaît à la sortie **TIW**.

Lorsque **TIW** est égal à **TSW**, la sortie **Q** est mise à 1.

Un front mentant sur l'entrée **R** ou **front descendant** sur l'entrée **IN** remet le temporisateur interne à zéro et la sortie Q passe à 0.

La description des différents paramètres ainsi que leurs type est résumée dans le tableau suivant :

Paramètre	Type de données	Description
IN	BOOL	Lancement de la temporisation
ZB	TIME	Base temps du retard
SW	INT	Facteur du retard
R	BOOL	Entrée RAZ

Description des paramètres de sortie :

Paramètre	Type de données	Signification
Q	BOOL	Sortie
TIW	INT	Temps interne (valeur actuelle de la tempo)
TSW	INT	Consigne de la temporisation au moment du front 0 > 1 sur IN

Tableau. II.9 Paramètres de la temporisation AKF_TE

Le chronogramme de cette temporisation est donné par la figure suivante :

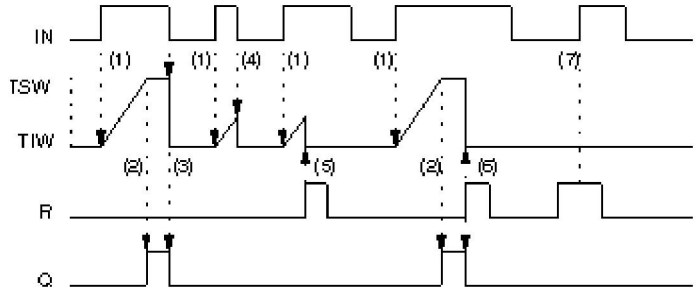
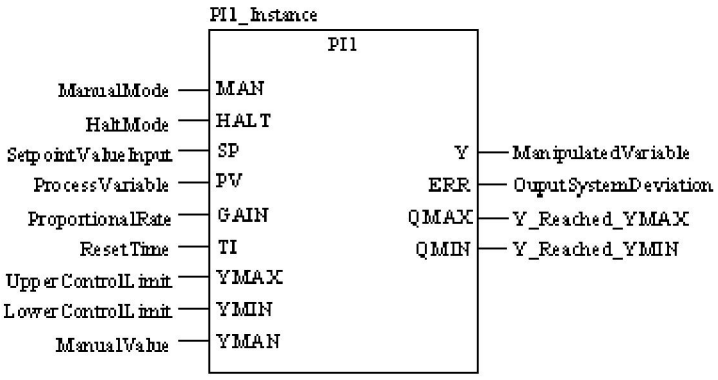


Fig. II.39 chronogramme de la temporisation AKF_TE

4.3.2. Le régulateur PI



Pour moduler l'éclairage de la rangée de luminaire en fonction de l'éclairage naturel (lumière du jour) nous aurons besoin d'un régulateur PI. On prendra le dénommé **PI1** dans Unity Pro.

Un écart du système **ERR** est constitué par la différence entre la consigne **SP** et la variable pilotée **PV**. Cet écart **ERR** entraîne la modification de la variable **Y** manipulée (commande).

Le bloc fonction a les caractéristiques suivantes :

- Modes de marche Manuel, Pause, Automatique
- Passage sans à-coup entre Manuel et Automatique
- Limitation de la variable manipulée

Sa fonction de transfert est donnée par : $G(s) = GAIN \times \left(1 + \frac{1}{TI \times s}\right)$

La description des paramètres de ce bloc est résumé par le tableau suivant :

Paramètre	Type de données	Description
MAN	BOOL	"1" : mode Manuel
HALT	BOOL	"1" : mode Pause
SP	REAL	Entrée de consigne
PV	REAL	Variable pilotée (mesure)
GAIN	REAL	Coefficient de l'action proportionnelle (gain)
TI	TIME	Temps d'action d'intégrale
YMAX	REAL	Limite supérieure
YMIN	REAL	Limite inférieure
YMAN	REAL	Valeur manuelle

Description des paramètres de sortie :

Paramètre	Type de données	Description
Y	REAL	Variable manipulée
ERR	REAL	Ecart du système de sortie
QMAX	BOOL	"1" = La sortie Y a atteint la limite supérieure
QMIN	BOOL	"1" = La sortie Y a atteint la limite inférieure

Tableau. II.10 Paramètres du régulateur PI utilisé

Le schéma fonctionnel de ce régulateur est illustré par la figure suivante :

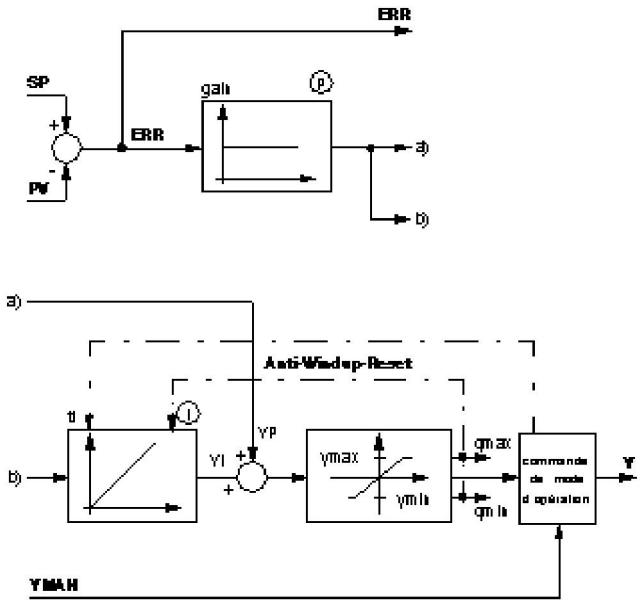


Fig. II.40 schéma fonctionnel du régulateur PI utilisé

Le programme en FBD pour cet exemple est donné en **Annexe E.**

4. CONCLUSION

Durant ce chapitre, nous avons vu que les automates programmables Schneider sont dotés d'un logiciel de programmation très performant qui est Unity Pro intégrant les cinq langages de programmation normalisés et possédant une interface graphique interactive et simple à utiliser.

Ce logiciel sera exploité par la suite pour la création des programmes de notre application.

CHAPITRE III

Introduction au Logiciel SCADA Vijeo Citect

1. Introduction

Vijeo Citect est un logiciel pour un système d'acquisition et de contrôle des données (SCADA ou Supervisory Control And Data Acquisition), ce logiciel permet de gérer et de surveiller des processus de fabrication, de distribution et de gestion d'installations.
Il est à noter que le logiciel Vijeo Citect est un logiciel polyvalent pouvant être utilisé aussi bien dans l'industrie que pour la Gestion Technique du Bâtiment, son utilisation en GTB permet de gérer l'ensemble de l'immeuble, surveiller la consommation d'énergie et les différentes alarmes, commander l'ouverture ou la fermeture des portes, commander l'éclairage, modifier la consigne de température pour les systèmes CVC…, enfin bref, tout ce qu'on peut imaginer.

2. Caractéristiques et avantages du logiciel Vijeo Citect

Le logiciel Vijeo Citect dispose de plusieurs caractéristiques faisant de lui une solution de choix pour la supervision des systèmes de GTB, parmi ces caractéristiques on peut citer les suivantes :

1. Analyse et présentation de tendances historiques et en temps réel sous forme de courbes et d'histogrammes (tel que la consommation d'énergie électrique des différents secteurs d'un bâtiment)
2. Gestion des alarmes (alarmes relatives aux systèmes de sécurité (anti-intrusion), alarmes déclenchées par des pannes d'équipements tel que les équipements de CVC), de plus le logiciel permet aussi la génération de rapports donnant l'état des équipements ainsi que les différents événements enregistrés (alarmes, pannes,…etc.).
3. Deux langages de programmation Cicode (langage évolué) et CitectVBA.
4. Eléments graphiques contenant Génie, Super Génie et objets ActiveX (ces éléments vont être explicités plus loin dans ce chapitre).
5. Possibilité d'utiliser des *grappes (ou Cluster)* permettant de regrouper plusieurs systèmes de contrôle en un seul système, ce qui permet de diminuer les couts, ainsi, si on utilise plusieurs systèmes de supervision pour des sites différents, on peut utiliser un Cluster pour les regrouper un une seule station permettant de superviser les différents sites et donnant ainsi la possibilité de comparer les données et tendances des différents sites.

3. Outils et composantes de Vijeo Citect

L'architecture Vijeo Citect peut être divisée en deux domaines de fonctionnalité :

1. configuration
2. exécution
3. pilotes

La configuration comprend toutes les tâches requises pour préparer et créer un projet tandis que l'exécution est l'implémentation d'un projet dans un environnement de production réel.

Pour décrire les outils fournis avec Vijeo Citect, le plus simple est d'examiner leurs rôles dans les phases de configuration et d'exécution.

3.1. Outils de configuration

Pour créer et configurer un projet sous Vijeo Citect, on utilise les outils suivants :

3.1.1. Explorateur Citect 

Application utilisée pour créer et gérer les projets Vijeo Citect. L'Explorateur Citect affiche une liste de tous les projets qui peuvent être renommé, sauvegardé ou supprimé.

3.1.2. Editeur de projet 

Cette application est utilisée pour créer et gérer les données de configuration relatif à un projet : tags (ou variables), les alarmes, éléments de système, éléments de communication (serveurs d'E/S, Cluster…) etc.

3.1.2. Editeur graphique Citect 

Application utilisée pour dessiner, créer et modifier les éléments graphiques d'un projet : modèles, objets graphiques, symboles, génies, super génies, etc. (la définition de ces éléments va être vue en détail plus loin), les éléments graphiques qu'on peut visualiser dans l'écran d'un superviseur sont créer grâce à cette application.

3.2. Outils d'exécution

Les outils suivants permettent d'exécuter, de surveiller et de piloter les projets en phase d'exécution, Vijeo Citect dispose des outils d'exécution suivants :

3.2.1. Vijeo Citect Client Web 

Cet outil permet d'afficher un projet Vijeo Citect dans un navigateur, cet outil permettra donc par exemple à un manager d'accéder aux ensembles de données relatives à un projet de

n'importe quel poste sur le réseau ayant des droits d'accès en lecture ou en lecture/écriture et ceci sans avoir recours à un logiciel supplémentaire.

3.2.2. Gestionnaire d'exécution Vijeo Citect 

Application utilisée pour gérer et piloter la configuration de l'UC du projet et l'état d'exécution de chaque élément.

3.2.3. Analyseur de processus 

Contrôle Active X permettant de comparer et d'analyser des tendances historiques en temps réel et des données d'alarme en phase d'exécution.

3.3. Les pilotes

Vijeo Citect peut communiquer avec toute une gamme de périphériques d'E/S, dont les automates, les contrôleurs de boucle et les systèmes de commande distribués.

Les pilotes permettent de communiquer avec des périphériques en utilisant divers protocoles de communication (Ethernet, TCP/IP, série, etc.) Le pilote définit les paramètres de projet requis pour que Vijeo Citect communique avec un périphérique donné. Ces paramètres concernent notamment les éléments suivants :

- Cartes
- Ports
- Périphériques
- Adressage des tags (variables)

4. Eléments d'un projet

Un projet sous Vijeo Citect se construit en utilisant plusieurs éléments, cette subdivision facilite la conception et la mise en œuvre des projets. Les éléments utilisés sont les suivants:

- Éléments graphiques
- Tags
- Alarmes
- Éléments système
- Éléments de communication
- Éléments des serveurs d'E/S
- Cicode / CitectVBA

Par la suite on reviendra en détail sur chacun de ces éléments.

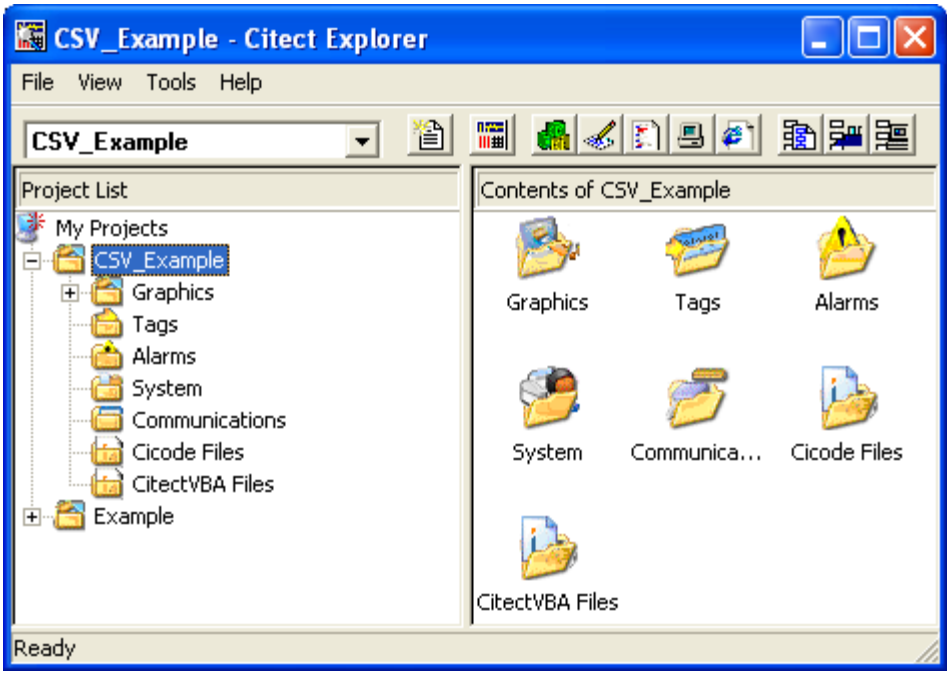


Fig. III.1 Eléments d'un projet sous Vijeo Citect

4.1. Éléments graphiques

Les éléments graphiques d'un projet Vijeo Citect sont utilisés pour créer les écrans affichés sur les clients de visualisation. Ils sont décrits ci-dessous (pour le moment en se contentera par des définitions et plus loin nous allons illustrer certaines notions par des exemples)

4.1.1. Les pages

C'est sur ces pages là qu'on peut créer et déposer tout les objets d'une application donnée.

4.1.2. Les pages modèles

Ce sont l'ensemble de pages utilisées pour standardiser les écrans, elles permettent donc de créer les modèles de pages relatifs à un projet donné et de les personnaliser selon le gout du concepteur.

4.1.3. Les symboles

Objets graphiques stockés dans une bibliothèque en vue de leur réutilisation.

4.1.4 Les génies

Objets groupant plusieurs éléments graphiques et fonctionnels pour faciliter leur duplication.

4.1.5. Les super génies

Génies pouvant transmettre des données spécifiques à un périphérique en phase d'exécution

4.2. Les tags (Variables)

Les tags permettent d'identifier les points de l'infrastructure à surveiller et à contrôler avec Vijeo Citect. Le nom donné à un tag devient une étiquette pour une adresse de registre, de sorte qu'il puisse être intuitivement appliqué aux pages graphiques. Un exemple de tag est la vitesse d'un ventilateur donné, la température au sein d'un bureau…etc.

4.3. Alarmes

Les alarmes permettent d'identifier des conditions dans un système Vijeo Citect nécessitant une intervention (panne d'un équipement donné, dépassement de valeur d'un tag donné…etc.)

4.4. Éléments système

Les éléments système d'un projet Vijeo Citect permettent de personnaliser, de gérer et de suivre le système superviseur. Ils comprennent :

4.4.1 Les commandes clavier

Ce sont des séquences de touches associées à des instructions, ainsi, ces commandes peuvent être utilisées comme raccourci pour des instructions répétitives.

4.4.2. Les rapport

Ils permettent de présenter l'état d'une installation et de donner des informations détaillées tel que l'état de certains équipements, la valeur de certaines variables (comme la température à l'intérieur des chambres, le niveau de consommation d'énergie dans certaines salles…etc.), ces rapports peuvent être donnés périodiquement ou suite à certain événement.

4.4.3. Les évènements

Ce sont des commandes exécutées en réponse à des déclencheurs tels que des expressions Cicode ou des tags de variables, par exemple, un évènement permet d'envoyer une commande d'ouverture de vannes de refroidissement lorsque la température dépasse une certaine valeur.

4.4.4. Les périphériques

Ce sont des éléments pouvant transférer des données de haut niveau à d'autres éléments tels que des fichiers RTF, des fichiers ASCII et des imprimantes.

4.4.5. Les projets inclus

Ce sont des projets prédéfinis contenant des configurations de base et qui sont automatiquement inclus dans les projets utilisateurs, l'utilisation de ces projets inclus permet au concepteur de projets de s'en passer de certaines détails de configuration et d'utiliser directement les configurations prédéfinies dans les projets inclus au lieu de prendre le temps de les redéfinir.

4.5. Éléments de communication

Les éléments de communication d'un projet Vijeo Citect correspondent à la représentation configurée du matériel de communication du système de supervision. Ils comprennent :

4.5.1. Les cartes

Composant permettant divers types de communication avec des périphériques d'E/S, elle peut faire office de serveur d'E/S.

4.5.2. Les ports

Connexion physique entre la carte et le périphérique d'E/S (comme le port COM d'un ordinateur)

4.5.3. Les périphériques d'E/S

Ce sont les appareils communicant avec le systèmes de supervision à l'instar des API ou des contrôleurs décentralisés.

4.6. Eléments de serveurs d'E/S

Les éléments des serveurs d'un projet Vijeo Citect correspondent à la représentation configurée des serveurs du système de supervision. Ils comprennent :

4.6.1. Le cluster

Le cluster (ou grappe) est un groupe logique de serveurs, ce regroupement permet de rendre ces serveurs accessibles à partir d'une seule station de supervision.

4.6.2. Les serveurs d'E/S

Serveurs de communication spécialisés échangeant des données entre les périphériques d'E/S et les clients de visualisation Vijeo Citect

4.6.3. Les serveurs d'alarmes

Serveurs surveillant les alarmes et les affichant sur le(s) client(s) de visualisation approprié(s).

4.6.3. Les serveurs de rapports

Serveurs contrôlant le traitement des rapports.

4.6.3. Les serveurs de tendances

Serveurs contrôlant l'accumulation et l'enregistrement d'informations de tendances.

Remarque : tous ces serveurs peuvent être des serveurs logiques définis sur une seule station.

5. Création et configuration d'un nouveau projet sous Vijeo Citect

Dans ce qui suit, nous allons illustrer par des exemples les principales notions liées à la création d'un nouveau projet, la configuration du matériel et l'utilisation des outils de VC.

5.1. Création d'un nouveau projet

Pour créer un nouveau projet sous Vijeo Citect, on doit suivre les étapes suivantes :

1. Lancez l'Explorateur Citect, l'éditeur graphique et l'éditeur de projet se lancent automatiquement.
2. Cliquez sur le bouton **Nouveau projet** ou, dans le menu **Fichier**,
3. Sélectionnez **Nouveau projet**.
4. Entrer un **Nom** pour le projet que vous désirez créer (obligatoire).
5. Entrer une **Description** ainsi que l'**Emplacement** où les fichiers du nouveau projet doivent être stockés.
6. Sélectionnez un **Style de modèles** et une **Résolution de modèles** pour définir l'apparence des pages graphiques.
7. Cliquez sur **OK** pour créer le projet ou sur **Annuler**.

Le nouveau projet et ainsi crée.

5.2. Configuration du nouveau projet

Notre projet étant crée, on doit maintenant le configurer en définissant le cluster, les différents serveurs et en exécutant l'assistant de configuration de l'ordinateur. Ces configurations permettent au superviseur de communiquer avec les périphériques, elles permettent aussi de définir des réglages de base tel que l'activation ou a désactivation des rapports, évènements, tendances…etc.

5.2.1. Définition de cluster et des serveurs

Comme on l'a déjà mentionné, un cluster correspond au regroupement d'un ensemble de serveurs les rendant accessibles à partir d'une seule station. Pour définir un cluster, on doit suivre les étapes suivantes :

1. Dans l'Éditeur de projets, Sélectionnez **Serveurs | Clusters**.
2. Entrez le nom du cluster et éventuellement un commentaire dans la boîte de dialogue Cluster come indiqué dans la figure ci-dessous.
3. Cliquez sur le bouton **Ajouter** pour ajouter un nouvel enregistrement ou sur **Remplacer** si vous avez modifié un enregistrement.



Fig. III.2 Définition d'un nouveau cluster

Le cluster étant défini, on peut à présent définir nos serveurs, pour cela, on doit suivre les étapes suivantes :

1. Sur l'éditeur de projet Citect, on sélectionne **Serveurs**
2. On choisit le serveur à définir (serveur d'E/S, d'alarmes, de tendances ou de rapport)
3. On entre les paramètres du serveur (le cluster associé, le nom du serveur, l'adresse sur le réseau (à laisser vide si le serveur est logique, i.e. défini au sein de la station de supervision)).
4. On clique sur le bouton **Ajouter pour** ajouter un nouveau serveur ou sur **Remplacer** pour effectuer une modification.

La figure V.3 illustre les différents champs permettant de définir un serveur est ses caractéristiques.

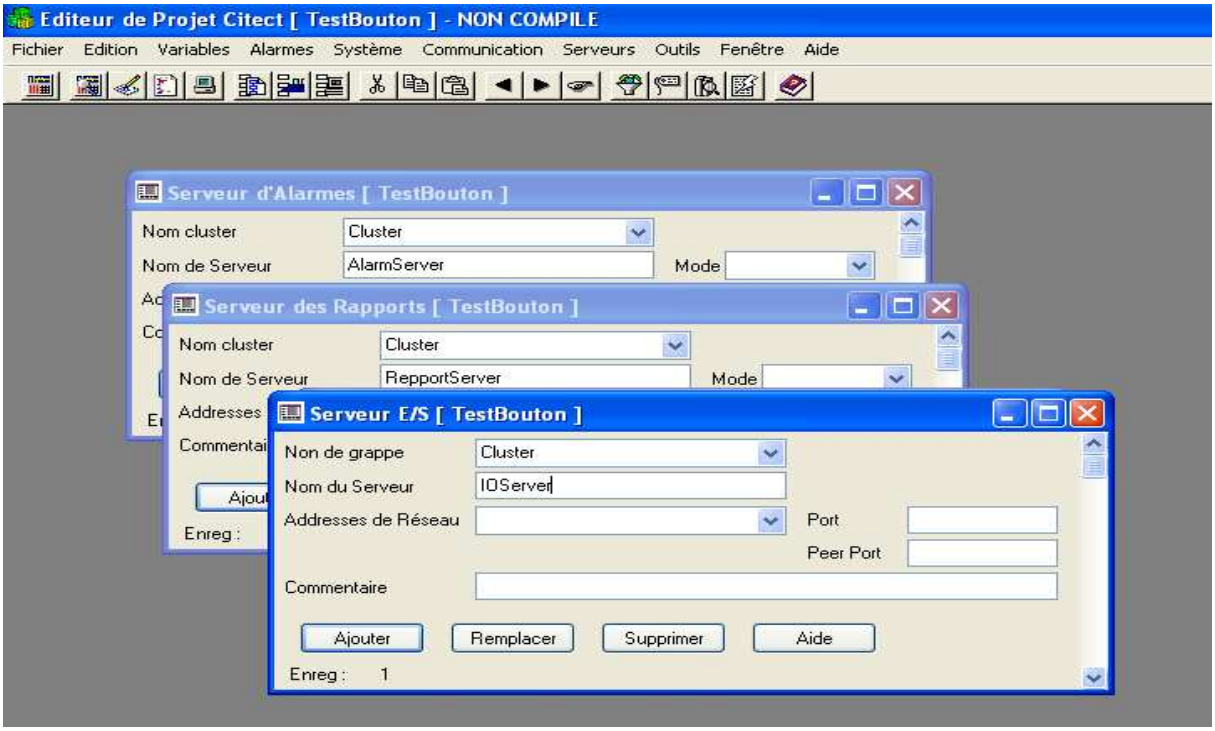


Fig. III.3. Définition des serveurs sur Vijeo Citect

5.2.2. Exécution de l'Assistant de Configuration de Poste

L'Assistant de Configuration de l'ordinateur permet de définir certains réglages de base tel que le rôle assigné à l'ordinateur, on effet, le poste de supervision peut être un serveur et un poste client à la fois pour les petits projets ne possédant pas des serveurs physiques, ou seulement un poste client (poste de visualisation) pour les projets où des serveurs dédiés (d'E/S, d'alarmes, de rapports,…etc.) ont été prévus.

Pour lancer l'Assistant de configuration Vijeo Citect, on doit suivre les étapes suivantes :

1. Ouvrir l'Explorateur Citect.
2. Dans la liste des projets, sélectionner **Mes Projets** - indiqué par une icône représentant un ordinateur.
3. Double-cliquer sur l'icône de l'Assistant de configuration de l'ordinateur ou sélectionner **Outils | Assistant** de configuration de l'ordinateur.
4. Sélectionner **Configuration express** ou **Configuration avancée**.
5. Suivre les étapes de configuration et choisir les options qui conviennent à notre projet (inhiber les alarmes ou non, duré de scrutation des alarmes…etc.)

En effectuant ces étapes, notre poste sera configuré et on peut désormais commencer à éditer notre projet et concevoir les pages constituant l'écran de supervision grâce à l'éditeur graphique Citect.

5.3. Configuration de la communication

Dans cette section, nous montrerons comment établir la communication entre le poste de supervision et les périphériques (calculateurs de terrain, automates, capteurs intelligents,…etc.), cette étapes est indispensable afin de permettre le transfert de données entre le superviseur et les périphériques.

Vijeo Citect dispose d'un outil intéressant permettant une configuration rapide de la communication, à savoir l'assistant Express Communications, cette assistant permet de créer un serveur d'E/S (ou de choisir un si le serveur a été déjà créer), il permet aussi de choisir le périphérique auquel on veut se connecter à partir d'une large gamme prédéfinie.

Les étapes suivantes illustrent comment utiliser l'assistant Express Communications :

1. Sur l'éditeur de projet Citect, on sélectionne le menu **Communication** puis **Assistant Express Communications**
2. On sélectionne le serveur d'E/S.
3. On sélectionne le périphérique d'E/S on le choisissant d'une liste s'il est déjà crée, sinon on entre le nom qu'on veut attribuer au nouveau périphérique.
4. On choisit le modèle de notre périphérique à partir d'une liste, comme indique par la figure V.4
5. On entre les différentes caractéristiques de notre périphérique ; adresse ip, protocole de communication (TCP ou UDP) ainsi que le port de communication utilisé (port COM pour une communication série)

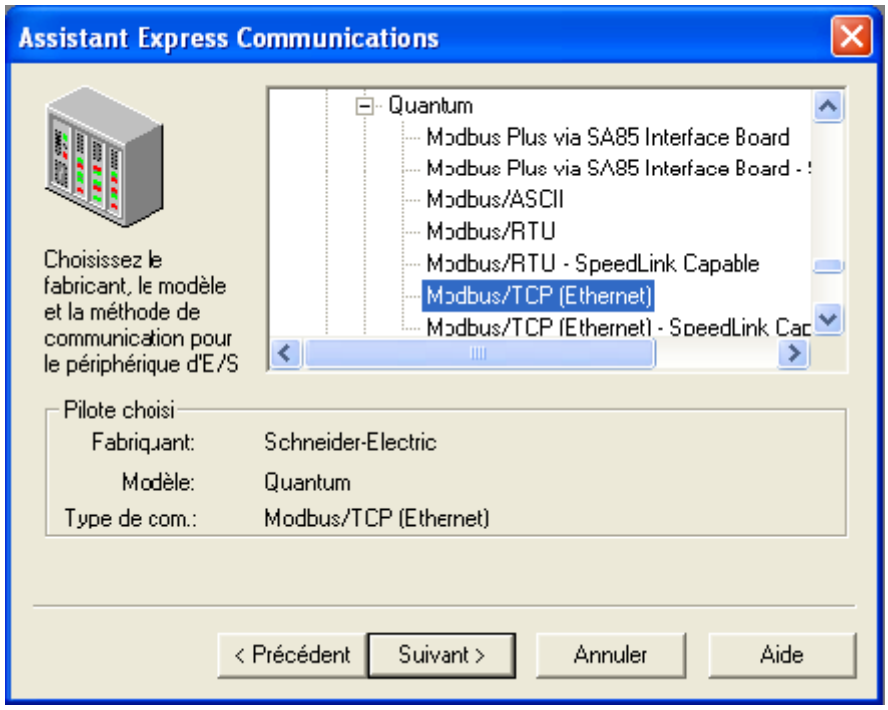


Fig. III.4. Choix du périphérique d'E/S.

Remarque : Afin d'établir une communication entre le Vijeo Citect et le simulateur de l'automate disponible au niveau du logiciel Unity Pro, on doit saisir l'adresse ip statique du

simulateur (adresse par défaut et figée) qui est : ***127.0.0.1*** comme indiqué par la figure ci-dessous

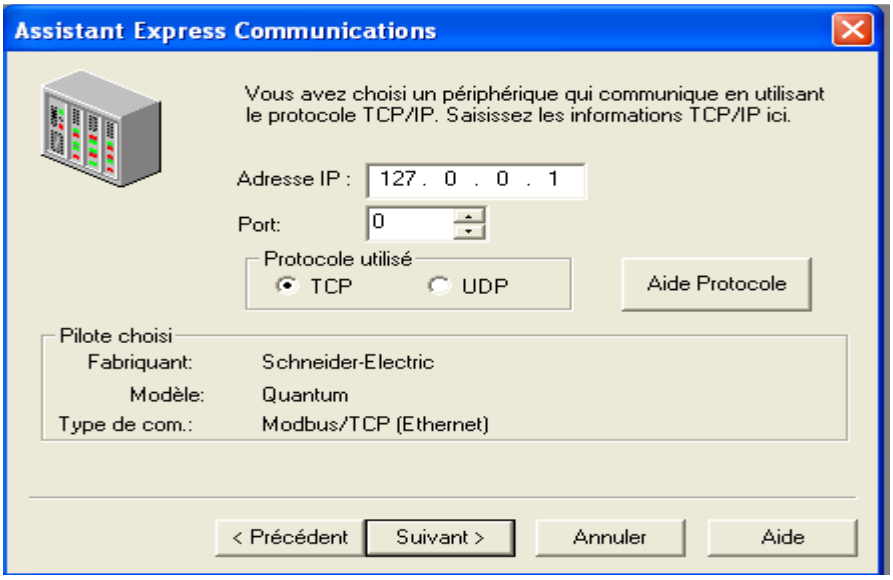


Fig. III.5. Configuration de l'adresse et du protocole de communication

6. Les graphiques

Nous allons maintenant entrer dans le vif du sujet et montrer comment créer les graphiques constituant un écran de supervision, pour cela, on considère une petite application à la base de laquelle on va illustrer tous les aspects en liaison avec la création des graphiques.

Cahier de charge : nous disposons d'un bureau muni de trois rangées de luminaires pour l'éclairage et de trois détecteurs de mouvement (chacun associé à une rangée) servant comme détecteur de présence, on veut que l'éclairage soit commandé par le détecteur de mouvement afin de l'éteindre lorsqu'aucune présence n'est enregistrée pour rendre ainsi la consommation d'énergie plus rationnelle.

- Une rangée de luminaire est allumée si le détecteur de mouvement associé détecte une présence, sinon elle est éteinte.
- De plus, le bureau dispose aussi de trois boutons poussoirs permettant l'allumage d'une rangée de luminaire même si le capteur de mouvement est désactivé (aucune présence n'est détectée), ces boutons peuvent être utile lorsqu'un occupant souhaite allumée d'autres rangées que celle sous laquelle il travaille dans le but d'augmenter la luminosité.
- L'allumage d'une rangée par un bouton poussoir sera activé pour une durée déterminée (20 mn), au bout de cette durée, les luminaires de la rangée vont s'éteindre dans le cas où le capteur de mouvement indique que les bureaux sous la rangée sont toujours vacants.

6.1. Création de la page principale

La figure suivante donne un aperçu de l'écran de visualisation de notre application sur la station de supervision.

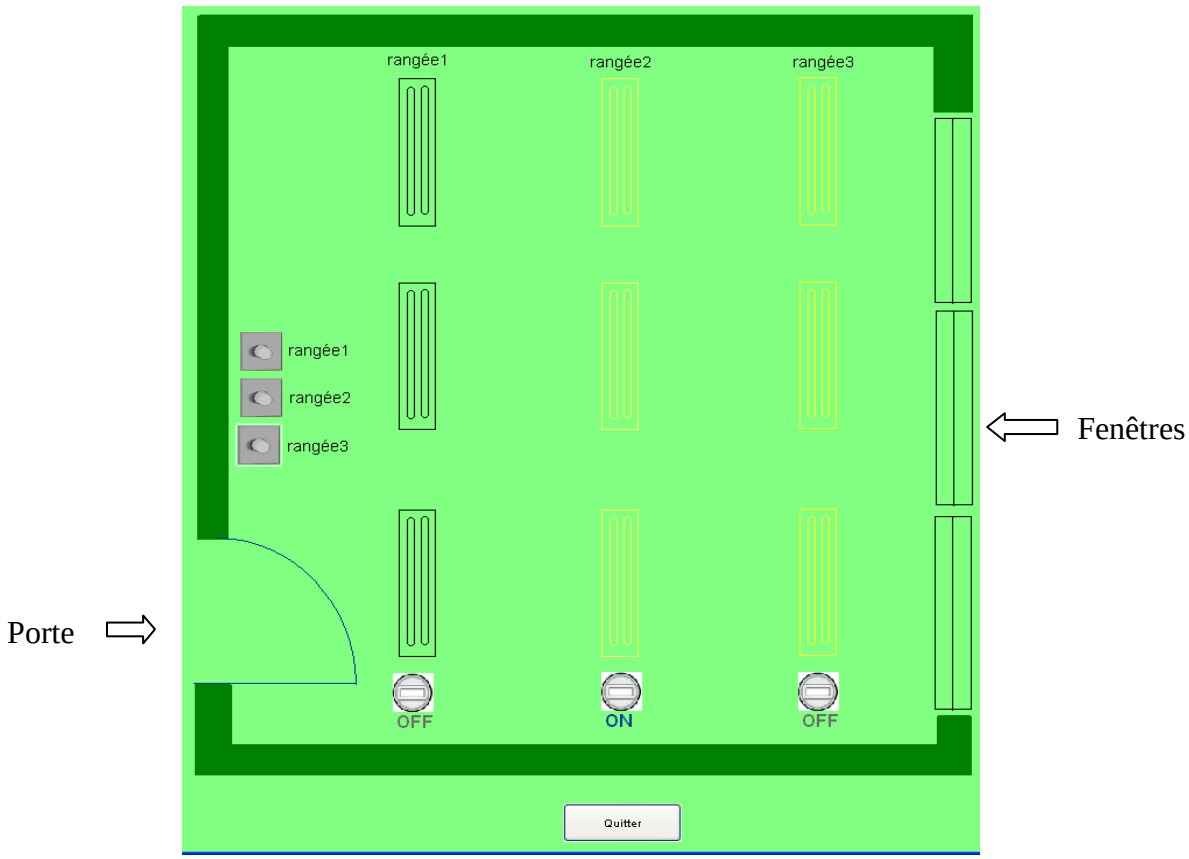


Fig. III.5. Ecran de visualisation de l'application

Sur cette figure, on voit que la deuxième rangée est allumée puisque le détecteur de mouvement est activé, par ailleurs, quoique le détecteur de mouvement de la troisième rangée est désactivé, on voit que les luminaires de cette rangée sont allumés et ceci suite à une action sur le bouton poussoir de la rangée 3.

Après avoir crée et configuré le projet en suivant toutes les étapes listées dans la section précédente, on ouvre une nouvelle page dans l'éditeur graphique Citect, cette page comportera tout les éléments graphiques de notre application.

A présent, on commencera par concevoir les éléments de base de notre application (le contour, les fenêtres, la porte et les luminaires), ces éléments sont à concevoir en utilisant les outils de dessein de Vijeo Citect tel que les polygones, le cercle, le rectangle…etc. cette étape étant simple et ne nécessite pas plus d'étalement.

La deuxième étape consiste à définir tout les tags auquel nous aurons besoin, ces variables permettent le transfert de données entre la station Vijeo Citect et le périphérique d'E/S.

6.1.1. Définition de tags sous Vijeo Citect

Chaque tag (variable) est défini par son nom, son type (digital, real, …etc.), le périphérique associé (qui doit être déjà défini lors de l'exécution de l'étape 'Assistant Express Communications') ainsi que son adresse sur le périphérique.

Nous allons à présent établir la liste de tous les tags à utiliser dans cette application.

Tout d'abord, on aura besoin de tags permettant de déterminer l'état d'un capteur de mouvement, l'information qui détermine l'état du tag provient évidement de l'automate auquel est connecté le capteur sur une carte d'entrée. Ces tags vont servir après pour déterminer l'expression à afficher au dessous du symbole du capteur ('ON' en bleu ou 'OFF' grisé).

De plus, on aura aussi besoin de trois tags servant à transmettre l'action d'appui sur les boutons poussoirs pour allumer les luminaires.

Ces six entrées vont servir à déterminer les trois sorties donnant l'état de chacune des rangées de luminaires, les sorties seront calculées grâce à un programme sur Unity Pro obtenu après traduction du cahier de charge en un grafcet.

Procédure pour définir un tag : Sur l'éditeur de projet Citect, on clique sur le menu tag, on choisit ensuite variables élémentaire, la fenêtre suivantes s'affichera alors

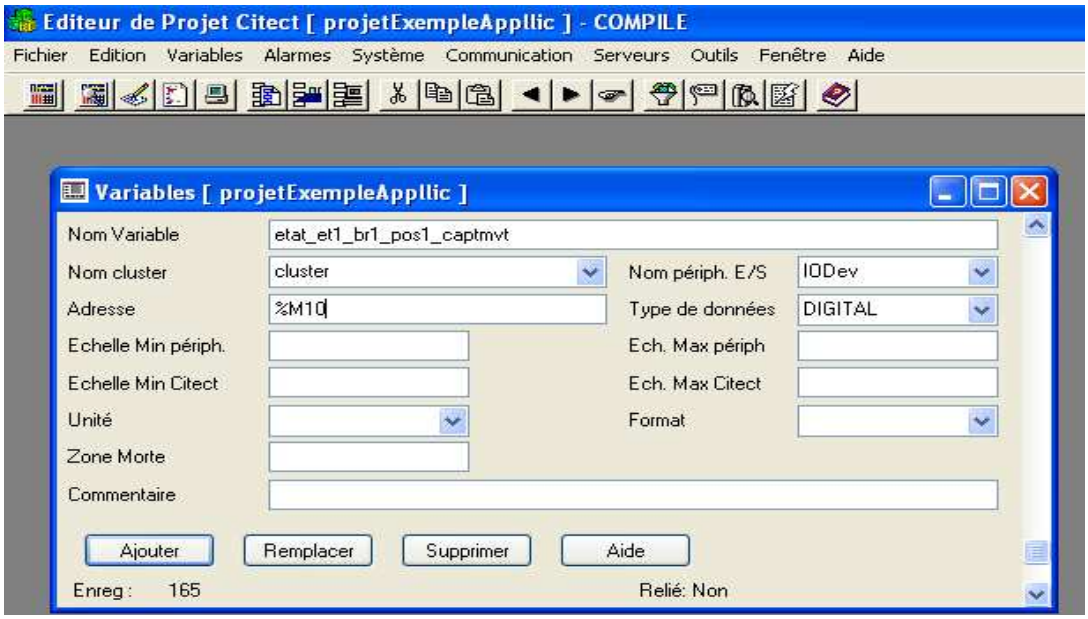


Fig. III.6 Définition de tags

La fenêtre qui s'affiche permet de saisir toutes les informations requise sur le tag :

- Le nom de la variable : ce nom doit être choisit de manière judicieuse et bien structuré, il est préférable qu'il comporte : le type de la variable (**état** d'un capteur (activé ou désactivé) ou bien **action** sur une sortie), l'emplacement et le type du périphérique relative à la variable.

- Le cluster associé au projet.
- Le nom du périphérique (l'automate).
- L'adresse de la variable sur le périphérique.
- Le type de données transférées par le tag (digitale, real…etc.)

Le tableau suivant donne tout les tags auquel on aura besoin pour cette application

Nom du tag	Type de données	Commentaire
etat_et1_br1_pos1_captmvt	DIGITAL	Tag donnant l'état du capteur de mouvement de la 1 rangée (postion 1)
etat_et1_br1_pos2_captmvt	DIGITAL	Tag donnant l'état du capteur de mouvement de la 2 rangée (postion 2)
etat_et1_br1_pos3_captmvt	DIGITAL	Tag donnant l'état du capteur de mouvement de la 3 rangée (postion 3)
action_et1_br1_pos1_BP	DIGITAL	Tag servant à transmettre l'action sur le Bouton Poussoir 1de la station de supervision vers le périphérique
action_et1_br1_pos2_BP	DIGITAL	Tag servant à transmettre l'action sur le Bouton Poussoir 2 de la station de supervision vers le périphérique
action_et1_br1_pos3_BP	DIGITAL	Tag servant à transmettre l'action sur le Bouton Poussoir 3 de la station de supervision vers le périphérique
etat_et1_br1_rg1	DIGITAL	Tag donnant l'état de la première rangée de luminaire (ON ou OFF)
etat_et1_br1_rg2	DIGITAL	Tag donnant l'état de la deuxième rangée de luminaire (ON ou OFF)
etat_et1_br1_rg3	DIGITAL	Tag donnant l'état de la troisième rangée de luminaire (ON ou OFF)

Tableau III.1 liste des tags utilisés

Remarque : concernant les noms donnés aux tags, et1 signifie étage 1, et br1 signifie bureau 1, en effet, on suppose que le bureau de notre application est le premier bureau localisé dans le premier étage d'un bâtiment, ainsi, notre petite application peut être insérée dans un projet contenant plusieurs étages et plusieurs bureaux et l'attribution de noms structurés aux tags permettra une extension plus facile.

Jusqu'à présent, nous avons crée les graphiques simples (la carcasse de notre projet : frontière du bureau, fenêtre, porte…etc.) ainsi que les tags auquel on aura besoin, la prochaine étape consistera à créer les génies relatifs capteurs de mouvement et aux boutons poussoir.

6.2. Création des génies et des super génies:

Un génie est un élément graphique permettant de combiner plusieurs éléments entre eux, de les associer à des tags, et d'ajouter le tout ensuite (le génie) à la bibliothèque, dans ce qui suit, nous allons illustrer les génies par des exemples.

Pour créer un nouveau génie, on doit aller sur l'éditeur graphique Citect, sélectionner le menu **Fichier** puis cliquer sur nouveau, la fenêtre ci-dessous s'affiche alors et on doit cliquer sur le bouton **Génie**, la fenêtre permettant la création du génie s'affiche alors.

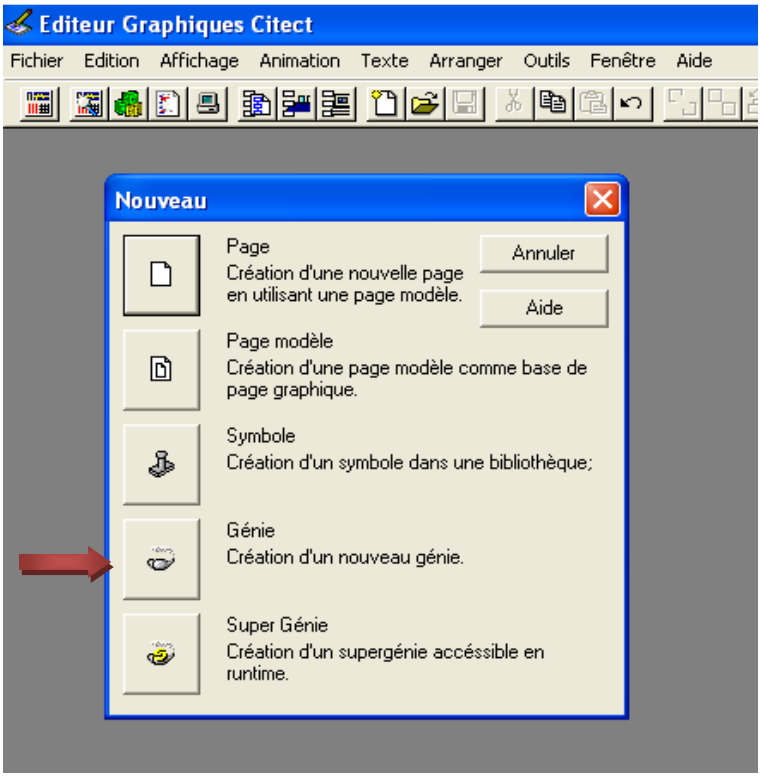


Fig. III.7 Création d'un nouveau génie

6.2.1. Création des génies des capteurs de mouvement

Après Avoir ouvert la page d'un nouveau génie en suivant les étapes décrit dans la section précédente, on commence par éditer le symbole du capteur, et si nous disposons déjà d'un symbole à partir d'un fichier externe, il suffit alors de coller l'image en faisant un clique droit sur la page du génie en en choisissons **coller**.

La prochaine étape maintenant (et la plus intéressante) consiste à programmer les deux symboles **ON** et **OFF** pour qu'ils s'affichent suivant l'état du tag '*etat_et1_br1_posI_captmvt*' où $I = \{1, 2, 3\}$

Pour éditer les deux textes **ON** et **OFF**, on doit cliquer sur l'outil TEXT (symbole **A**) dans la barre d'outils sur la page du nouveau génie et ceci comme indiqué par la figure V.8, on positionne ensuite le curseur à l'endroit où le texte doit s'afficher, **on appui sur la touche espace**, ensuite on fait un double clique, la fenêtre des propriétés du texte s'affiche alors, on commencera par l'édition de l'apparence du texte comme indiqué sur la figure V.9, pour cela, on clique sur l'onglet **Affichage Valeur** indiqué par la flèche, on suite on édite le texte VRAI

et le texte FAUX et on fait entré le tag déterminant quel valeur le texte va prendre (le texte VRAI s'affiche si le tag choisi possède la valeur1)

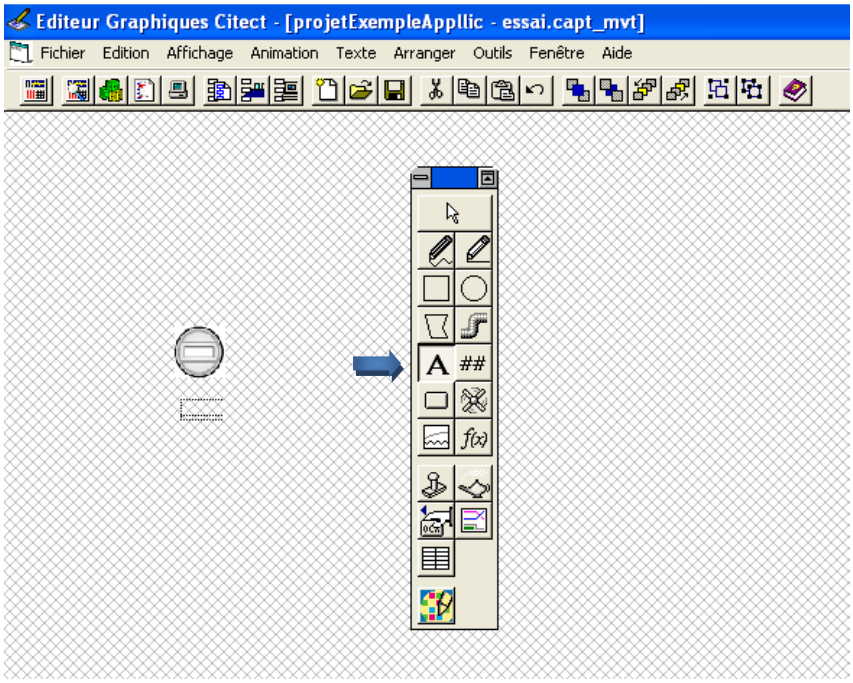


Fig. III.8. Ajout d'un texte sous le symbole du capteur

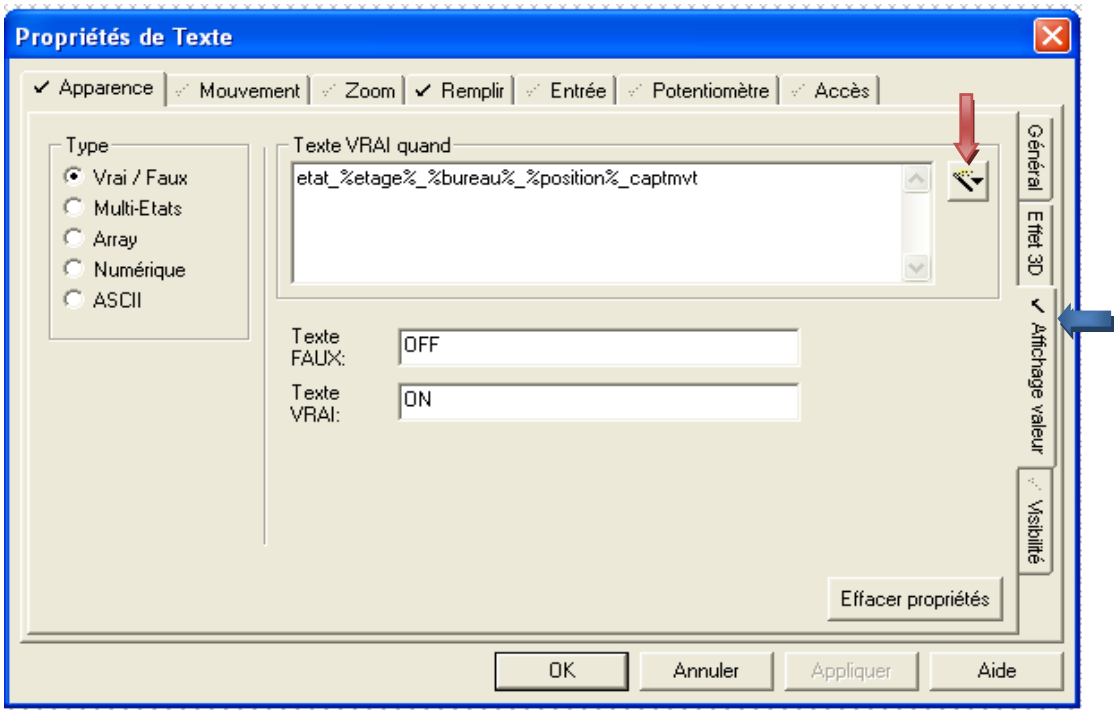


Fig. III.9. Edition de l'apparence du texte

Choix du tag associé au texte : deux possibilités existent

1. Soit on choisit un tag on cliquant sur le bouton indiqué par la flèche rouge descendante, la liste de tous les tags crées s'affiche alors et on peut sélectionner le tag

désiré. Cette option n'est pas intéressante car on impose ainsi des tags *statiques* et le génie crée ne peut alors être utilisé avec d'autres tag rendant ainsi son option de duplication inutilisable.

2. Soit utiliser des chaines de substitution ''%'' comme indiqué dans la figure V.9, cela permet une attribution *dynamique* de tags et ceci lors de son appel dans la page principale, plus concrètement, si on fait comme dans l'exemple en écrivant ''etat_%etage%_%bureau%_%position%_captmvt'' dans la case *Texte Vrai quand*, et après enregistrement du génie dans la bibliothèque, si on colle le génie dans la page principale (on expliquera plus loin comment sauvegarder un génie et comment l'appeler de la page principale) , un **popup** s'affiche et on peut alors entrer le tag associé au génie, la figure V.10 illustre le popup et les champs permettant d'entrer le tag, on voit donc que tout ce qui est entre les délimiteurs de substitution '%' sera affiché dans le popup et prendra la valeur entrée (etat_%etage%_%bureau%_%position%_captmvt sera donc remplacé par état_et1_br_1_pos2_captmvt)

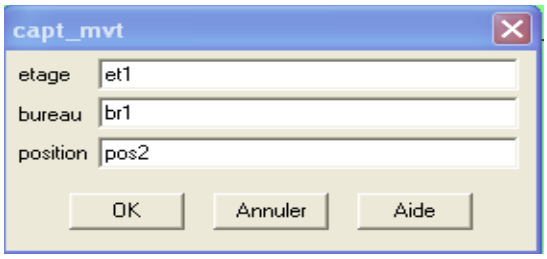


Fig. III.10. popup permettant d'associer un tag à un génie

Jusqu'à présent, nous avons créer un génie comportant le symbole d'un capteur et un texte ayant deux valeurs (ON et OFF) et on a associé des tags *dynamiques* permettant de déterminer quelle valeur le texte prendra, Nous allons maintenant développer un peu plus notre génie on associant des couleurs au texte qui s'affiche ; pour cela, on clique sur l'onglet remplir puis sur couleur comme indiqué par les flèches sur la figure

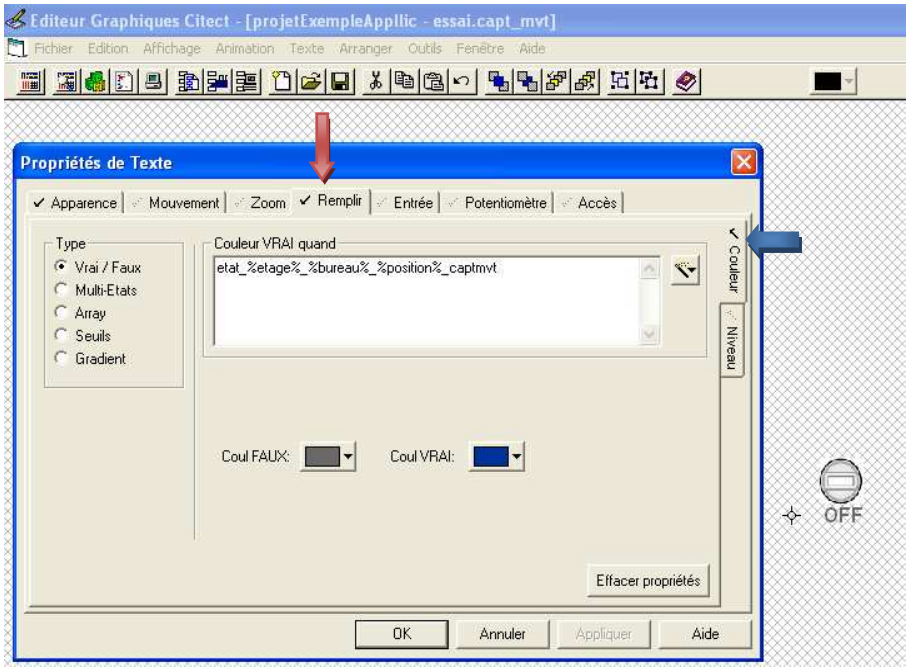


Fig. III.11. Programmation de la couleur du texte

Ainsi donc, suivant la valeur du tag associé au génie, le texte prend la couleur bleu pour ON ou gris pour OFF.

Ceci termine donc la conception de notre génie, reste à montrer comment le sauvegarder et l'appeler ensuite de la page principale.

Pour sauvegarder un génie, il suffit d'aller sur le menu **Fichier**| **Sauvegarder sous**, la fenêtre ci-dessous s'affiche alors, on peut choisir le nom du génie, la librairie (bibliothèque) dans laquelle il va être sauvegarder, et le projet auquel il appartient.

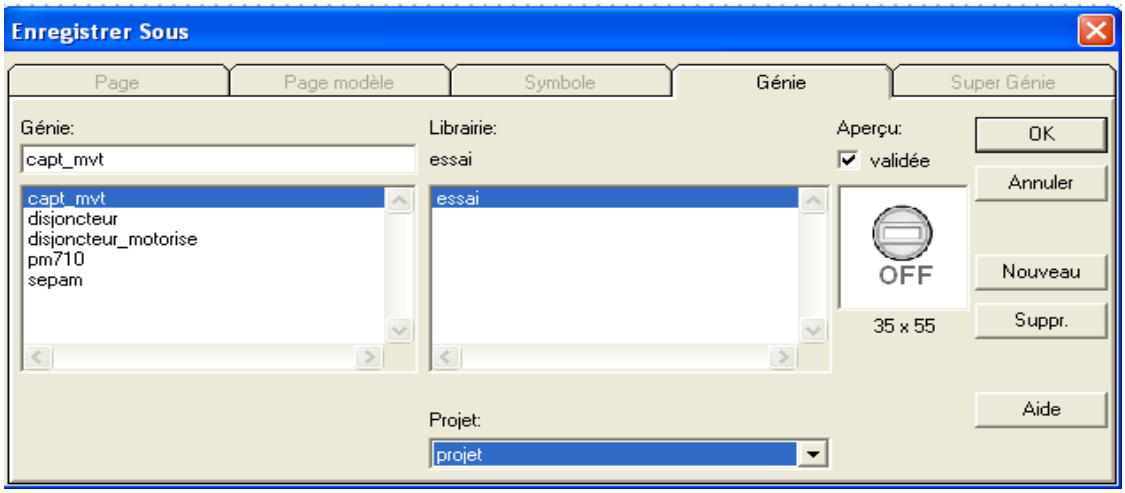


Fig. III.11. Sauvegarde d'un génie, association à une librairie et à un projet

Coller le génie dans la page principale : pour ce faire, on sélectionne l'outil **génie** (indiqué par la flèche bleue) dans la barre d'outils de la page principale, on sélectionne ensuite la librairie à laquelle le génie appartient (essai) on choisit ensuite le génie désiré parmi les génies

appartenant à cette librairie, cela étant fait, le popup déjà discuté s'affichera et nous permettra de lier des tags à notre génie.

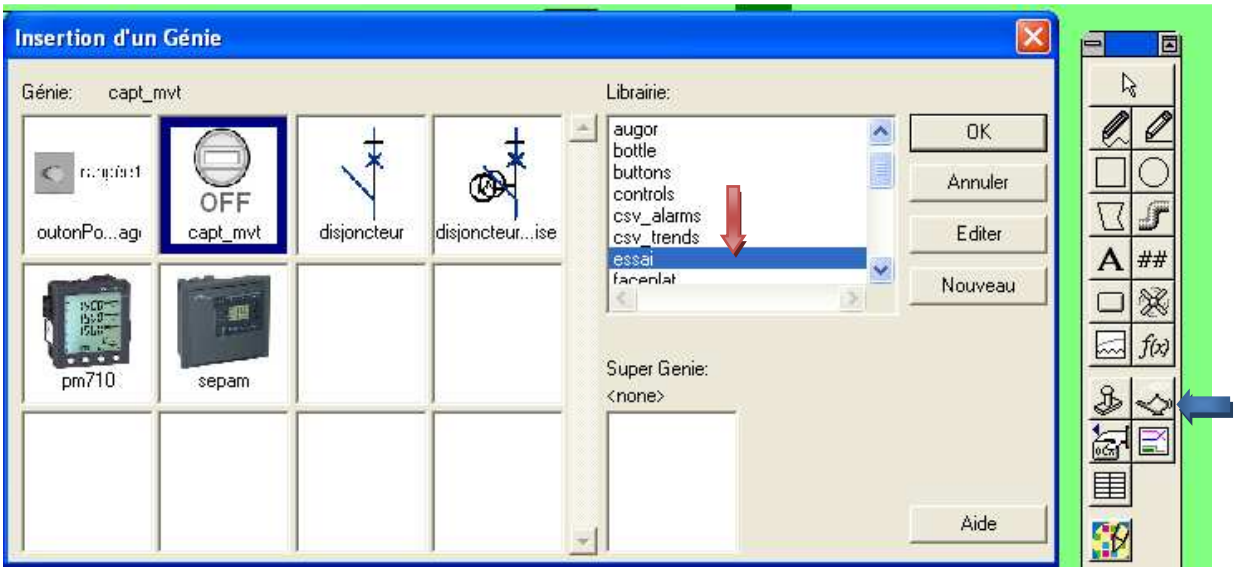


Fig. III.12. Coller un génie dans la page principale

6.2.2. Création des génies et des supers génies des boutons poussoirs

Comme on l'a déjà mentionné, les boutons poussoirs permettent de commander manuellement les luminaires et de les allumer (pour un temps déterminé) même en cas de non présence de personnes. Nous allons maintenant créer un génie comportant le symbole du bouton poussoir, en cliquant sur ce génie, on veut faire apparaitre un popup menu (le super génie) comportant un bouton qui permet l'allumage d'une rangée de luminaire. Tout cela va s'éclaircir étape par étape.

Commençons par créer un nouveau génie (comme indiqué sur la figure V.7), sur la page du génie, on insère le symbole du bouton poussoir (image bitmap) ramené d'un fichier externe, pour configurer le génie de façon à faire apparaitre le super génie après un clique, on utilise la fonction AssPopUP, comme indiqué sur la figure V.13, on clique donc d'abord sur **Entrée** puis **Clique** (fenêtre Propriétés du bouton, comme indiqué par els flèches), on active la case **Haut**, puis on introduit la fonction AssPopUp ainsi que ses arguments.

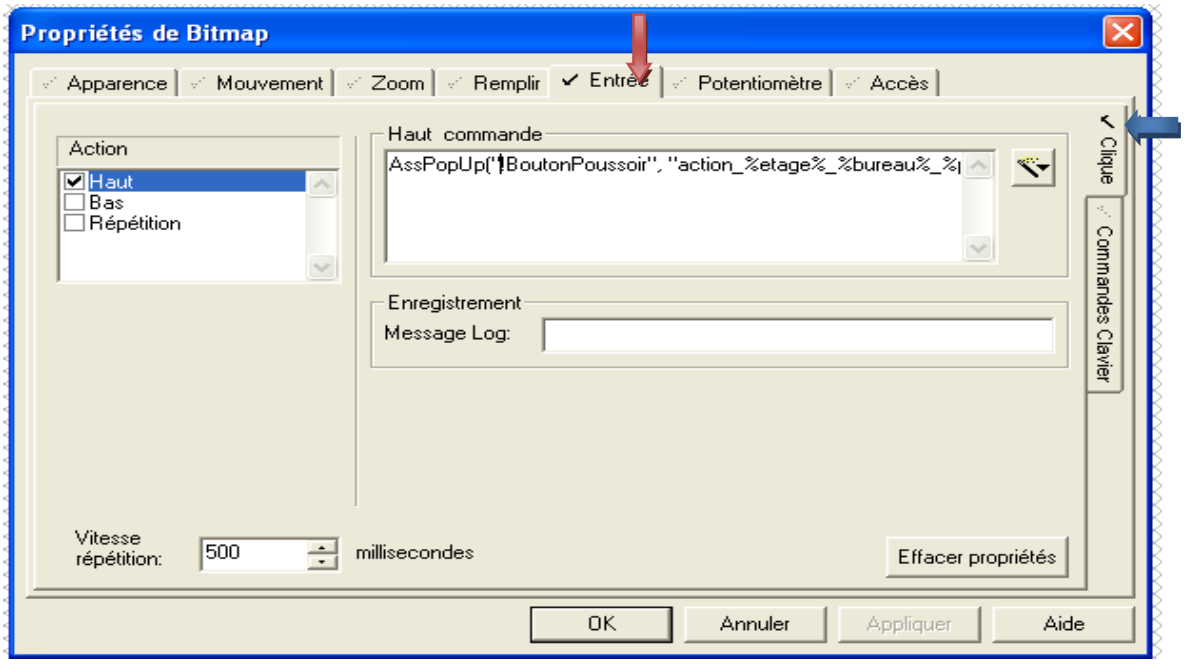


Fig. III.13 Configuration du génie

La fonction AssPopUp admet comme arguments le nom du super génie à associé au génie ainsi que tous les tags que le génie doit transmettre au super génie. Sur le champs de commande, on doit introduire la fonction AssPopUp avec les arguments suivants : *AssPopUp("!BoutonPoussoir", "action_%etage%_%bureau%_%position%_BP")*, où :

- !BoutonPoussoir représente le nom du super génie qu'on va créer après.
- *action_%etage%_%bureau%_%position%_BP* sont les tags que le génie doit transmettre au super génie.

Création du super génie : le super génie est un pop menu comportant en général des boutons de commande et qu'on associe à un génie. Pour créer un nouveau super génie, on doit aller sur l'éditeur graphique Citect, sélectionner **nouveau** dans le menu **Fichier**, puis sélectionner **super génie**, ceci étant fait, la page du super génie s'ouvre et on peut commencer sa configuration. On commence d'abord par ajouter un bouton qui permettra l'allumage de la rangée de luminaire (le bouton doit être ajouté dans la zone délimitée par un trait discontinu), ceci étant fait, on configure le bouton maintenant afin qu'il puisse changer l'état du tag passé par le génie et par suite allumer le luminaire.

Pour configuré le bouton, on édite ses propriétés et on clique sur **Entrée** puis **Clique** comme indiqué par les flèche, ensuite dans le champ de commande, on écrit le code illustré sur la figure ; à savoir : **?1 ?=1 ; Sleep(10) ; ?1 ?=0.**

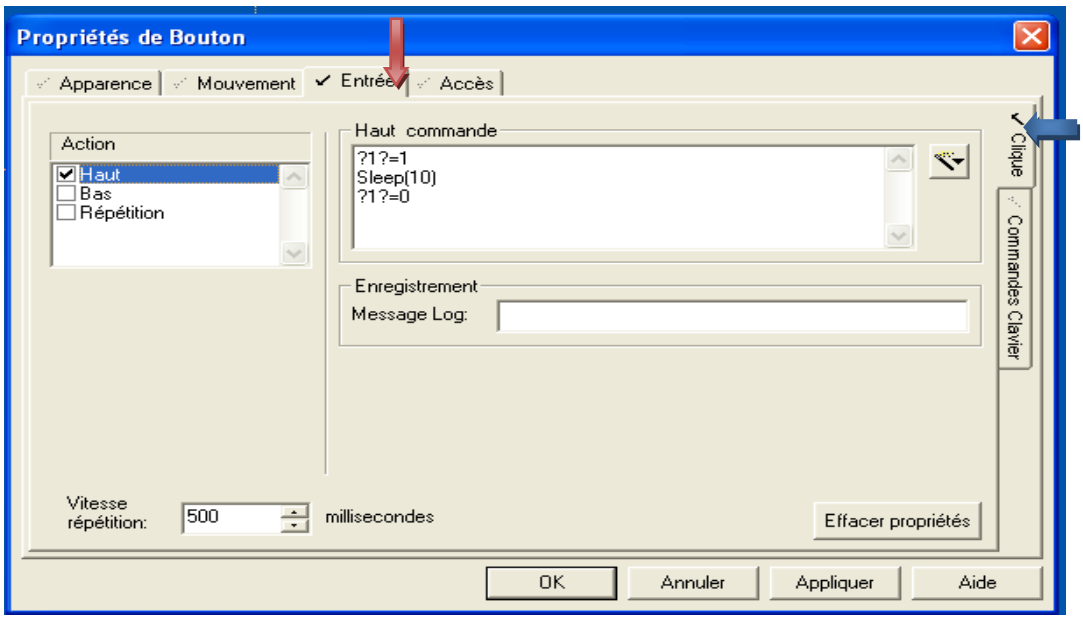


Fig. III.14. Configuration du bouton d'allumage dans le super génie

Explication du code : ?1?=1 signifie que le premier tag passé par le génie à son super génie grâce à la fonction AssPopUp (dans notre cas le tag est : *action_%etage%_%bureau%_%position%_BP*) va être mis à 1 , ainsi donc, le numéro qui est entre points d'interrogation indique le numéro du tag passé par le génie à son super génie.

Dans le cahier de charge, on a mentionné que l'allumage des luminaires par un bouton poussoir doit être temporisé afin d'éviter le risque d'oublie d'un luminaire allumé, ceci justifie donc l'ajout de la fonction **Sleep**, qui doit permettre un allumage de 20mn comme le cahier de charge l'indique (ici on se contente à 10s pour des besoins de simulation) cette temporisation passé, on remit le tag à 0 pour éteindre le luminaire en écrivant **?1 ?=0 ;**

La configuration du bouton d'allumage est ainsi terminée, on doit à présent ajouter un autre bouton permettant de quitter la page du super génie lorsque celle-ci est appelé, après ajout du bouton, on édite ses propriétés, on clique sur **Apparence,** puis **Symbole**, puis sur **Config**, ensuite on choisit le symbole approprié de la liste comme indiqué sur la figure.

L'étape suivante est d'ajouter la fonction permettant de quitter le super génie, pour cela, dans les propriétés du bouton, on va sur **Entrée** ensuite on ajoute la fonction **WinFree()** dans le champ de commande comme indiqué par la **figure III.14.**

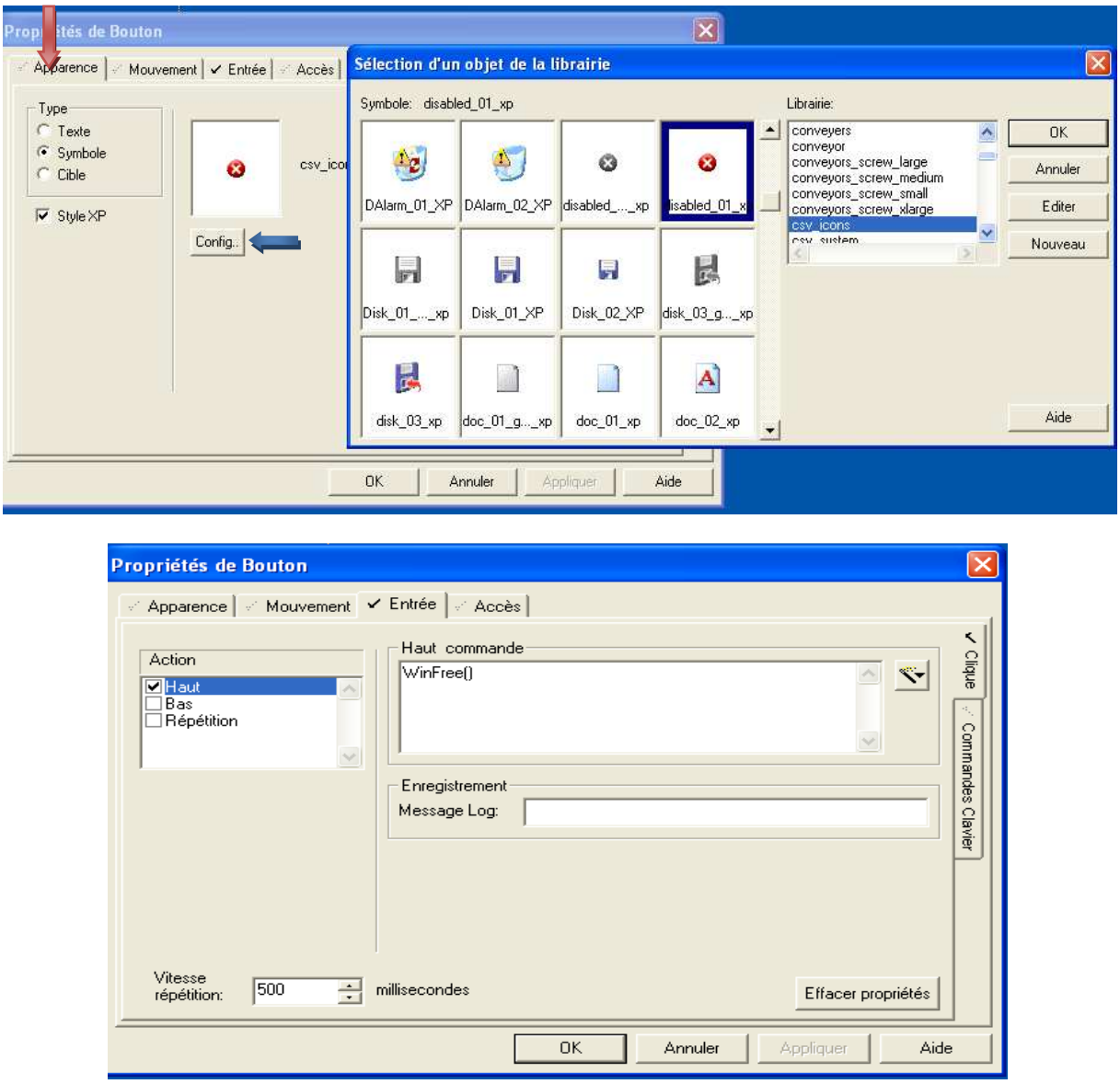


Fig. III.15. Configuration du bouton quitter dans le super génie

Ceci termine donc la configuration de notre super génie, reste à le sauvegarder sous le nom !BoutonPoussoir.

Notre génie étant crée, on doit maintenant lier le génie du bouton poussoir au super génie, pour cela on ouvre le génie, on clique sur **Edition, Lier des super génies**, on ajoute ensuite le super génie comme indiqué sur la **figure III.16.**

La **figure III.17** donne un aperçu de l'exécution de notre application où le super génie apparait suite à un clique sur le génie correspondant ; le bouton poussoir. Le clique sur le bouton **Allumer** du super génie permet d'allumer la deuxième rangée malgré le fait que le détecteur de mouvement est désactivé.

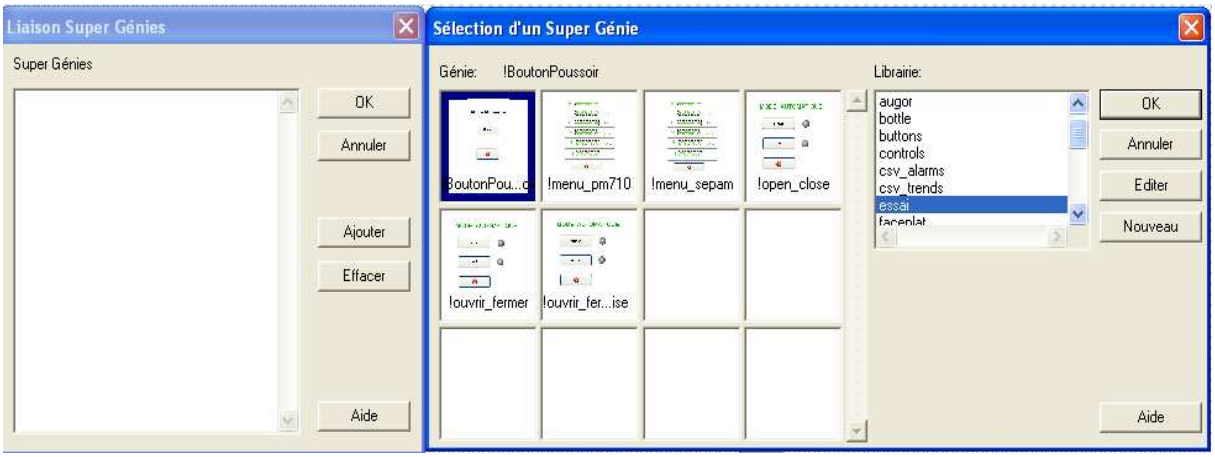


Fig. III.16. Lier un super génie à un génie

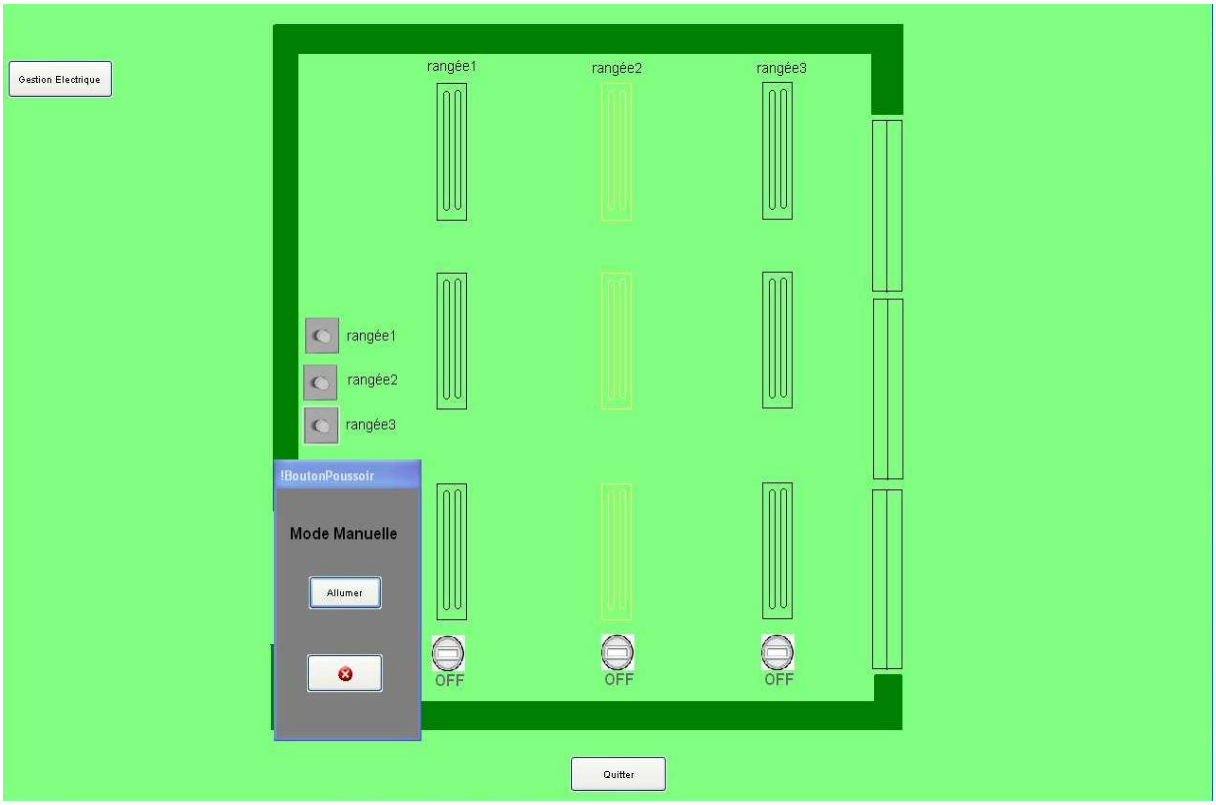


Fig. III.17. Aperçu de l'exécution de l'application, appel d'un super génie

Remarque : pour configurer le luminaire afin qu'il change de couleur lorsqu'il est allumé, il suffit d'éditer ses propriétés, de cliquer sur **Remplir** puis d'ajouter le tag approprié comme indiqué sur la figure, le luminaire devient ainsi jaune lorsque le tag prend la valeur 1

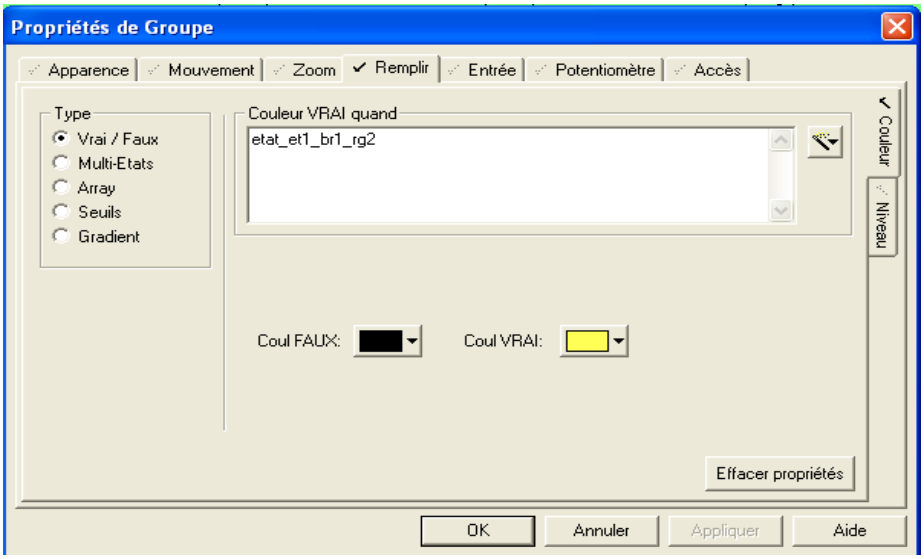


Fig. III.18. Configuration d'une rangée de luminaire

7. Les évènements

Les événements servent à déclencher une action, comme une commande ou une série de commandes. Ils permettent par exemple d'indiquer l'achèvement d'un processus à un opérateur ou de déclencher l'exécution d'une série d'instructions lorsqu'un processus franchit une étape spécifique.

L'activation d'un évènement peut se faire par trois méthodes :

1. Automatiquement ; à un temps donné et avec une périodicité définie.
2. Automatiquement ; lorsqu'un déclencheur prend la valeur vraie.
3. Automatiquement ; lorsqu'un déclencheur prend la valeur vraie et un temps donné avec une périodicité définie.

Exemple d'évènement : Nous allons maintenant illustrer les évènements par un exemple, en se référant à notre exemple d'application précédemment défini, on considère maintenant un autre scénario, on veut maintenant adopter une gestion horaire de l'éclairage de notre bureau ; les luminaires s'allument à une heure donnée (8 :00h par exemple) et s'éteignent à une heure donnée (18 :00h par exemple) pour cela il suffit de définir deux évènements et de leurs associer les actions adéquates. La définition de ces deux évènements se fait en suivant les étapes suivantes :

Etape 1 : Sur l'éditeur de projet Citect, on clique sur **Systèmes**, on sélectionne ensuite **évènements**, une fenêtre apparait alors permettant de configurer l'évènement.

Etape 2 : on édite les propriétés de l'évènement comme indiqué sur la figure V.18, on commence par choisir le nom de l'évènement, **Heure** permet de définir l'heure de

déclenchement de l'évènement, et **Période** définie la périodicité de l'évènement (dans notre cas 24h, puisque on veut que l'allumage soit quotidien), dans le champs **Déclenchement**, on choisit (éventuellement) le tag permettant de déclencher l'évènement (notre évènement sera déclenché indépendamment d'une condition, donc pas besoin de tag déclencheur), dans le champs **Action**, on écrit le code définissant toutes les actions relatives à notre évènement (ici c'est la mise à 1 des tags relatifs à l'éclairage afin d'allumer les luminaires des rangées)

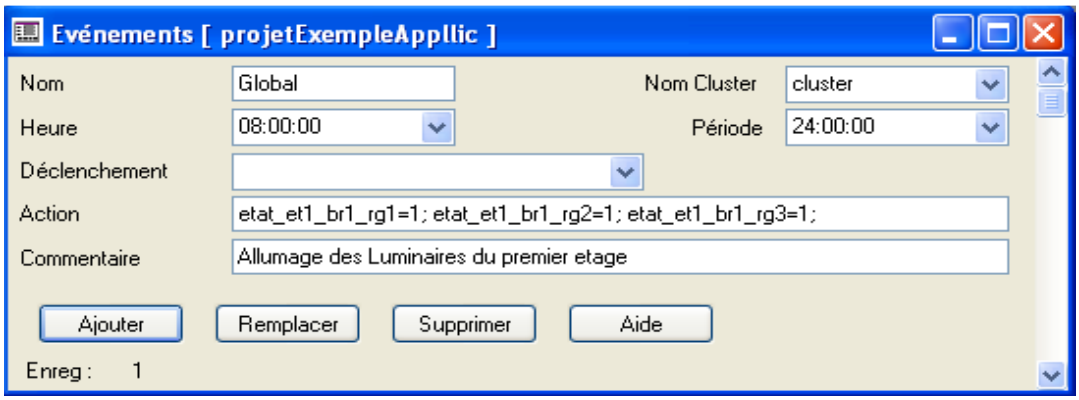


Fig. III.19. Configuration d'un évènement

Etape 3 : notre évènement étant maintenant défini, reste à l'activer en utilisant l'assistant de configuration de poste (mode avancé), cette étape n'est pas nécessaire si on choisit Global comme nom de l'évènement puisque tous les évènements ayant comme nom global seront activés par défaut.

8. Les alarmes

Les alarmes permettent d'alerter d'opérateur d'évènements spécifiques, il existe deux types d'alarmes :

1. **Les alarmes matérielles** : Vijeo Citect exécutent automatiquement des programmes de diagnostique pour vérifier l'état des périphériques, les défauts éventuelles sont reporter automatiquement à l'opérateur, cette option est intégrée automatiquement à Vijeo Citect et ne nécessite pas de configuration.
2. **Les alarmes configurées** : au contraste des alarmes matérielles, ces alarmes sont à configurer par le développeur pour permettre de **signaler** des évènements spécifiques.

Nous allons maintenant illustrer comment configurer des alarmes de base en se basant sur un petit exemple d'application.

8.1 Exemple d'application

En considérant toujours notre application précédente, on veut maintenant configurer une alarme pour signaler l'intrusion d'une personne, cette personne sera active si l'un des détecteur de mouvement détecte la présence d'une personne entre 20 :00 h du soir et 8 :00h du

matin, en cas de déclenchement d'alarme, l'opérateur pourra alors vérifier (à l'aide d'une camera de surveillance par exemple) qu'il s'agit bien d'un intrus et non pas de quelqu'un du personnel.

8.1.1. Etape 1 : vérification de la plage horaire

Pour configurer notre évènement, on a besoin d'abord de vérifier qu'on est entre 20 :00h et 8 :00h, pour cela la méthode la plus simple est de mettre un nouveau tag à 1 en utilisant un évènement et ceci comme décrit dans la section 7. La figure V.20 illustre les deux évènements permettant de marquer l'intervalle dans lequel on veut signaler une intrusion.

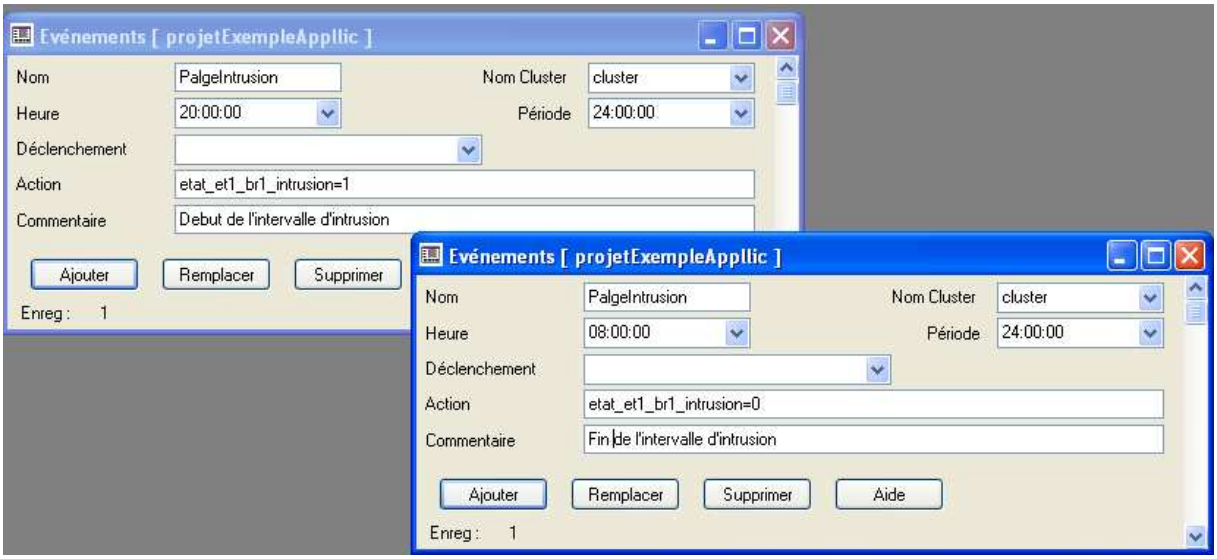


Fig. III.20. Configuration de deux évènements afin de marquer l'intervalle de detection d'intrusion

8.1.2. Etape 2 : Configuration de l'alarme

Nous allons maintenant configurer notre alarme, pour cela, on commencera d'abord par définir un serveur d'alarme (s'il n'est pas déjà défini), pour ce faire, il suffit de suivre les étapes expliquées dans la section **5.2.1**

L'étape suivante consiste à configurer l'alarme proprement dite, sur l'éditeur de projet Citect on clique sur le menu alarme, on choisit alors **Alarmes Digitales,** la fenêtre illustrée sur la figure V.21 apparait alors et permet d'éditer la configuration de l'alarme.

Sur la fenêtre on voit donc les propriétés suivantes :

Tag de l'alarme : Nom de l'alarme. Le nom doit être unique au cluster.

Nom du cluster : Nom du cluster qui exécute cette alarme. Si le nom du cluster n'est pas défini, Vijeo Citect estime que l'alarme est exécutée sur tous les clusters définis.

Nom de l'alarme : Nom du périphérique physique associé à l'alarme. Cette propriété est facultative. Vijeo Citect ne l'utilise que pour afficher les détails d'une alarme à l'écran ou pour les enregistrer dans un périphérique.

Desc. Alarme : Description de l'alarme. Elle peut inclure des données variables. Cette propriété est facultative. Vijeo Citect ne l'utilise que pour afficher les détails d'une alarme à l'écran ou pour les enregistrer dans un périphérique.

Tag A / Tag B : Les variables numériques (tags) déclenchant l'alarme, l'alarme devient active lorsque le les deux tags (ou seulement le tag A si le champ du tag B est laissé vide) prennent la valeur 1.

Catégorie Numéro ou étiquette de la catégorie d'alarme. Cette propriété est facultative. Si la catégorie n'est pas spécifiée, l'alarme utilise par défaut la catégorie 0.

Délai (hh:mm:ss) Délai de l'alarme ; Une alarme horodatée devient active lorsque l'état de la condition de déclenchement reste vrai pendant l'intervalle de temps défini par ce paramètre.

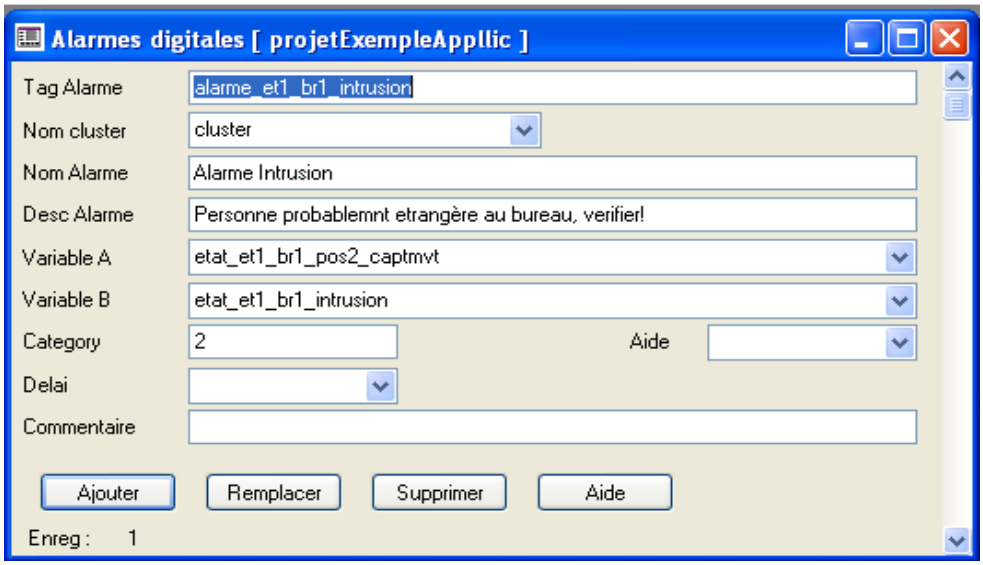


Fig. III.21. Configuration d'une alarme digitale

Sur la figure V.21, on a configuré une alarme digitale relative au capteur de mouvement de la deuxième rangée, on refait la même chose ensuite avec les deux autres capteurs (en entrant comme variable A les deux tags etat_et1_br1_pos1_captmvt et etat_et1_br1_pos3_captmvt successivement

Sur la figure, on remarque qu'on a utilisé la catégorie 2, l'étape suivante consistera à éditer ses propriétés

8.1.2. Etape 3 : Configuration de la catégorie d'alarme

Une catégorie permet de définir les caractéristiques de l'affichage (police, type de page), de l'enregistrement (imprimante ou fichier), et l'action à entreprendre lors du déclenchement d'une alarme (par exemple, activer une alarme sonore).

Pour configurer une catégorie, on sélectionne **Alarmes** puis **Catégorie d'Alarmes.** Une fenêtre apparait alors (Fig. V.22) permettant d'éditer les propriétés de cette catégorie.

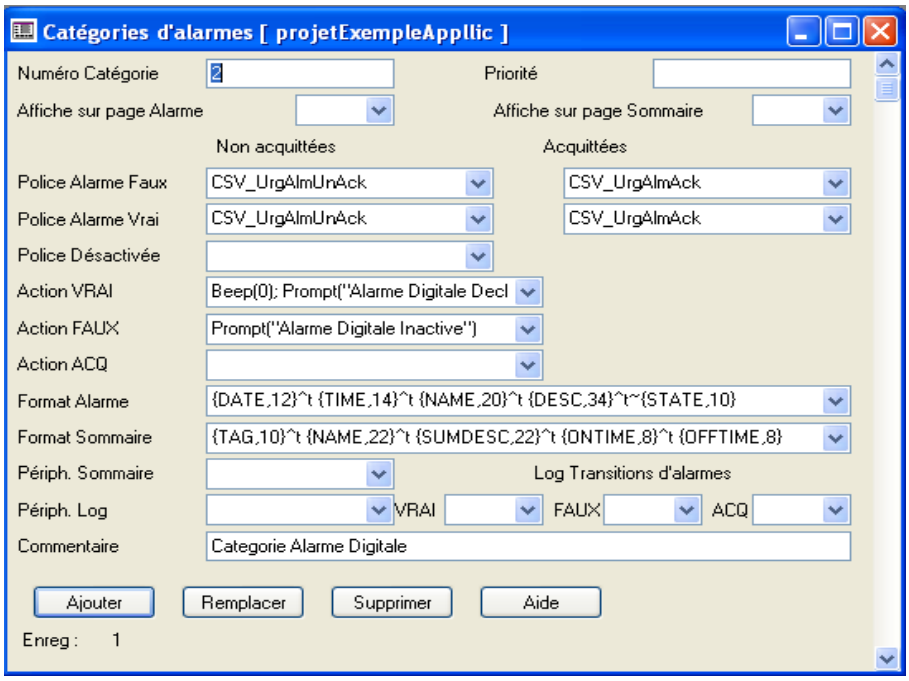


Fig. III.22 Configuration d'une catégorie d'alarme

Les propriétés d'une catégorie sont expliquées ci-dessous :

Numéro de catégorie : Catégorie de l'alarme (0-16 375) la catégorie 254 est réservée aux entrées de l'historique d'alarmes créées par l'utilisateur. La catégorie 255 est réservée aux alarmes matérielles.

Affichage sur la page d'alarmes : Détermine si les alarmes de cette catégorie sont affichées sur la page d'alarmes. La valeur par défaut de ce champ est VRAI.

Affichage sur la page d'historique : Détermine si les alarmes de cette catégorie sont affichées sur la page d'historique. La valeur par défaut de ce champ est VRAI.

Non acquittées : police des alarmes désactivées : Police utilisée pour l'affichage des alarmes qui ne sont plus actives (mais n'ont pas été acquittées). Cette propriété est facultative.

Acquittées : police des alarmes désactivées : Police utilisée pour l'affichage des alarmes qui ne sont plus actives et qui ont été acquittées. Cette propriété est facultative.

Non acquittées : police des polices actives : Police utilisée pour l'affichage d'une Alarme active non acquittée. Cette propriété est facultative. Si vous ne précisez pas de police, la valeur par défaut est 10 pt JAUNE.

Acquittées : police des alarmes actives : Police utilisée pour l'affichage des alarmes actives acquittées. Cette propriété est facultative. Si vous ne précisez pas de police, la valeur par défaut est 10 pt CYAN.

Action ON : Commande Cicode exécutée lorsqu'une alarme de cette catégorie est acquittée. Exemple : ***Action Alarme ON*** : STOP_PROCESS = 1; La variable numérique STOP_PROCESS est activée (ON) lorsqu'une alarme de cette catégorie est déclenchée.

Action OFF : Commande Cicode exécutée lorsqu'une alarme de cette catégorie est réinitialisée (254 caractères maximum). Exemple : ***Action Alarme OFF:*** ENABLE_PROCESS = 1;

La variable numérique ENABLE_PROCESS est activée (ON) lorsqu'une alarme de cette catégorie est réinitialisée.

Action ACK : Commande Cicode exécutée lorsqu'une alarme de cette catégorie est acquittée.

Périphérique d'historique : Périphérique auquel l'historique d'alarme est envoyé. Une alarme est enregistrée dans le périphérique d'historique lorsqu'elle est désactivée après avoir été affichée pendant une période supérieure à celle spécifiée par le paramètre SummaryTimeout. Une alarme enregistrée dans ce périphérique est supprimée de la page d'historique. Cette propriété est facultative. Si vous ne la spécifiez pas, les historiques d'alarmes ne sont pas enregistrés.

Périphérique d'enregistrement : Périphérique auquel les changements d'état de l'alarme sont envoyés. Une alarme est écrite dans ce périphérique à chaque fois qu'elle change d'état (ex. on, off, acquittée, activée, désactivée). Lorsque l'alarme est imprimée ou écrite dans un fichier ou un périphérique, le format spécifié dans ce périphérique prime sur le format d'affichage. Cette propriété est facultative. Si vous ne la spécifiez pas, les changements d'états ne sont pas enregistrés.

Enregistrement des transitions d'alarmes ON : Enregistre les détails des alarmes lorsqu'elles deviennent actives. La valeur par défaut de ce champ est VRAI.

Enregistrement des transitions d'alarmes OFF : Enregistre les détails des alarmes lorsqu'elles deviennent inactives. La valeur par défaut de ce champ est VRAI.

Enregistrement des transitions d'alarmes ACK : Enregistre les détails des alarmes lorsqu'elles sont acquittées (6 caractères maximum). La valeur par défaut de ce champ est VRAI.

8.1.3. Exécution de l'exemple

Afin d'afficher une alarme lors de l'exécution de notre application, on change le modèle de page de notre application, on adopte un modèle préconfiguré contenant déjà le raccourci permettant d'aller vers les pages d'alarmes, pour cela on sélectionne **Fichier** puis **Nouveau,** on sélectionne après le modèle de page **XpStyle ; Normale**

Ceci étant fait, on compile ensuite le projet et on l'exécute, la page illustrée sur la figure V.23 s'affiche alors. En allant sur Unity Pro, on peut mettre les tags relatifs aux capteurs de présence à 1 pour simuler une présence. Si on se trouve dans la plage d'heure où le programme de detection d'intrusion est activé, on entendra alors un Bip et le symbole d'alarme (flèche rouge) se mettra à clignoter alors. Dans le bas de la page s'affichera aussi le nom de l'alarme et l'heure de son déclenchement.

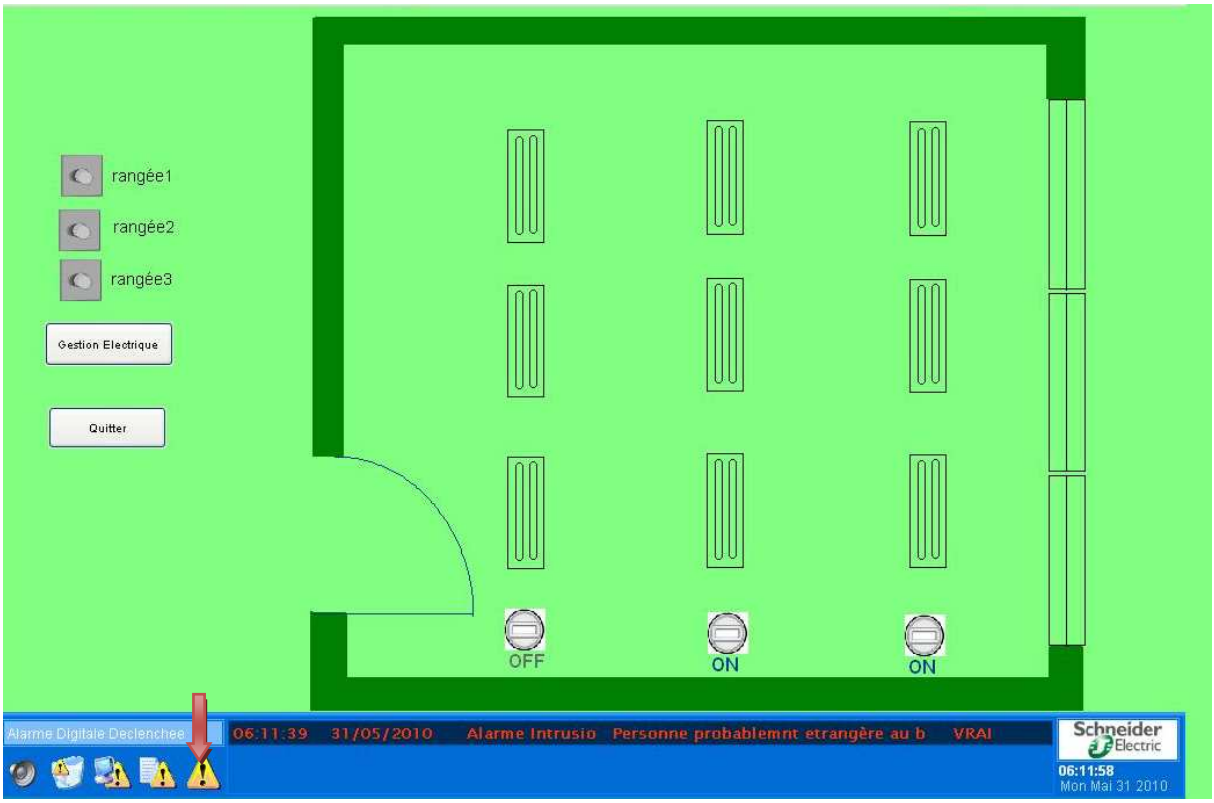


Fig. III.23 Exécution de l'application et affichage de l'alarme

En sélectionnant le symbole de la page d'alarme (flèche rouge), la page d'alarme s'affichera (Fig. V.24), et on peut alors visualiser toutes les informations relatives à l'alarme.

Noter que certaines informations (On Time, Duration…etc.) ne seront affichées que si les périphériques d'historique et d'enregistrement ont été définis.

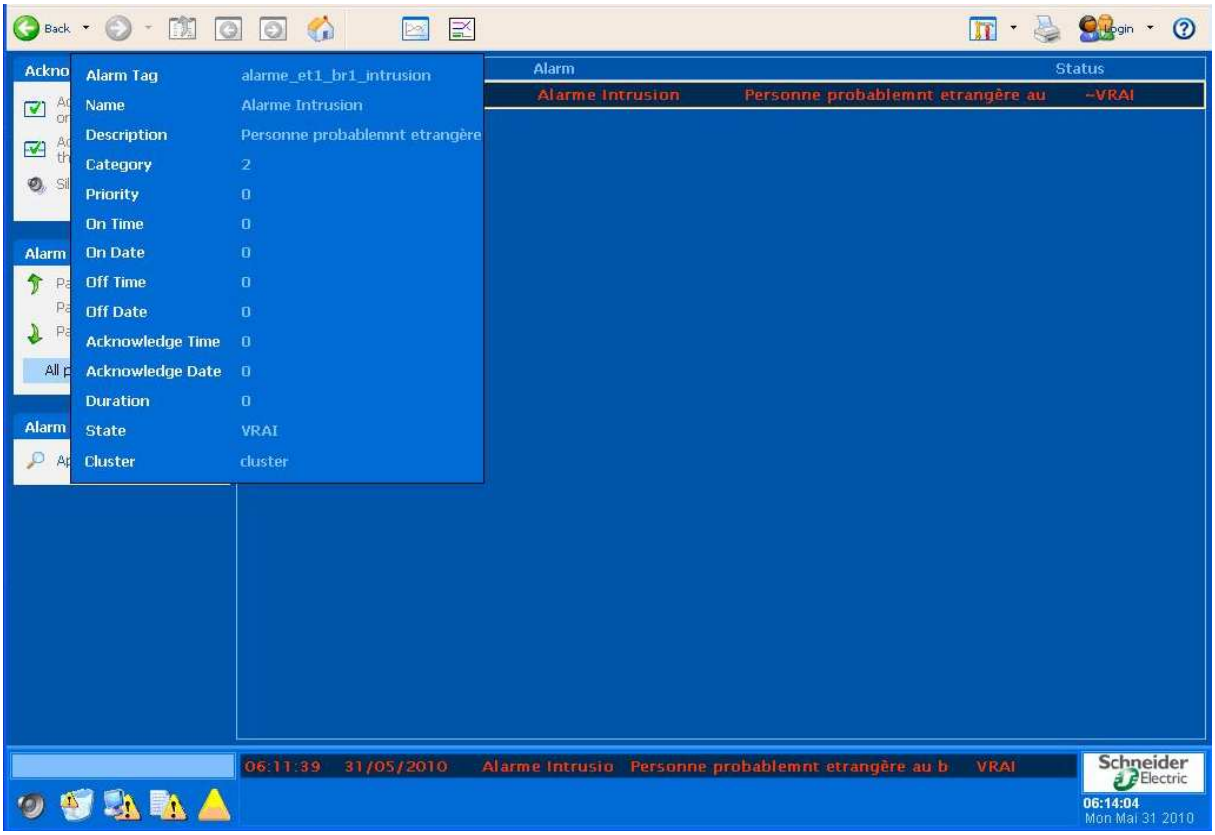


Fig. III.24. Page d'alarme

9. Les rapports

Les rapports permettent d'avoir des informations sur l'état de l'installation ou sur des conditions particulières de fonctionnement. Ces rapports peuvent être générés à la demande, à des heures spécifiées ou lorsque certains événements se produisent (comme le changement d'état d'un bit d'adresse). La sortie des rapports est contrôlée par un périphérique. Les rapports peuvent être imprimés lorsqu'ils sont générés ou sauvegardés sur disque pour une impression ultérieure. Ils peuvent être affichés, édités ou imprimés à l'aide d'un éditeur de texte ou d'un logiciel de traitement de texte, ou affichés sur une page dans Vijeo Citect. Ils peuvent également inclure des instructions Cicode qui s'exécutent au moment de leur production.

9.1. Configuration d'un rapport

Pour concevoir, configurer et utiliser un rapport on doit suivre les étapes suivantes :

9.1.1. Définition d'un périphérique de sortie pour le rapport

Un périphérique (périphérique Soft et non pas *Hard*) permet de transférer des données de haut niveau (un rapport, une commande, un journal d'alarmes, etc.) entre Vijeo Citect et d'autres éléments (une imprimante, une base de données, un fichier RTF, un fichier ASCII, etc.) du système Vijeo Citect. Les périphériques sont semblables aux périphériques d'E/S en ce qu'ils

permettent à Vijeo Citect d'échanger des données avec d'autres parties du système de contrôle et de supervision.

Le périphérique de sortie pour le rapport permettra de stocker le rapport dans un fichier, Pour le définir, on va sur **Systèmes** puis on sélectionne **Périphériques**, une fenêtre apparait alors (**figure III.25**.) et permet de configurer le périphérique.

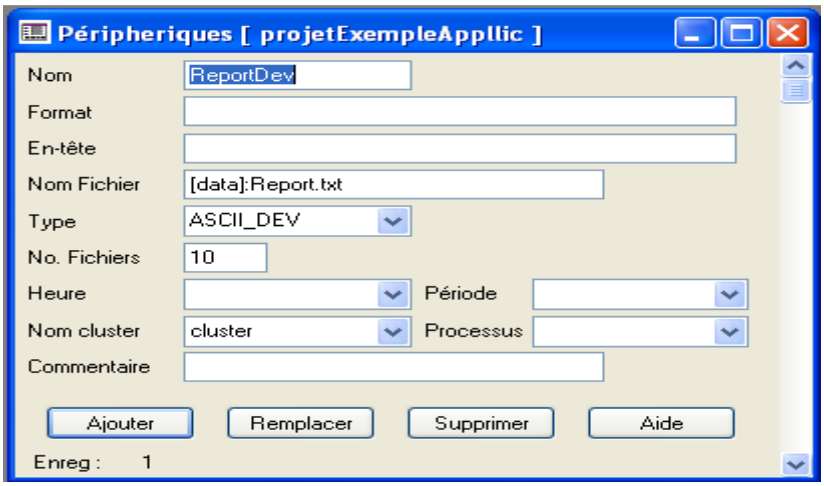


Fig. III.25. Configuration d'un périphérique de sortie pour le rapport

Pour une explication sur les différents champs de la fenêtre, veuillez consulter le help.

9.1.2. Configuration des propriétés des rapports.

Après avoir défini le périphérique, l'étape suivante consiste à configurer les propriétés du rapport, pour cela on sélectionne **Système** puis **Rapports,** la fenêtre ci-dessous (**Figure. V.26**.) s'affiche alors

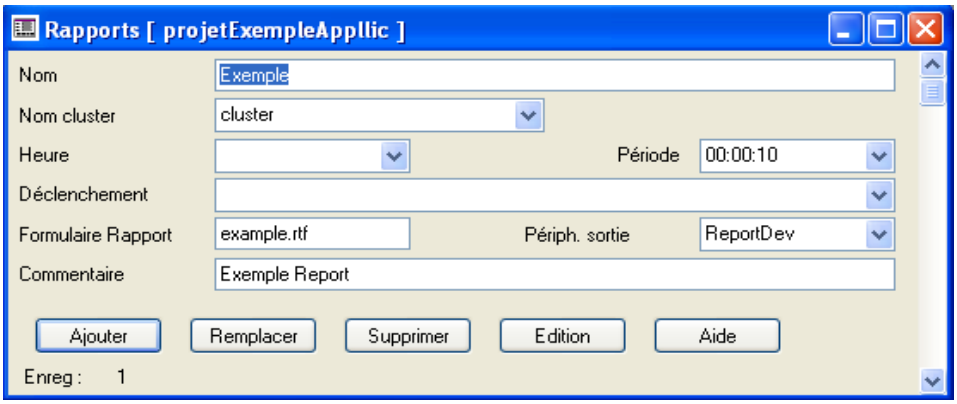


Fig. III.26. Configuration des propriétés d'un rapport

On peut éditer alors les propriétés de notre rapport. Dans ce qui suit, on présente de façon succincte les différents champs de la fenêtre précédente.

Nom du rapport : Le nom attribué au rapport, il doit être unique au cluster.

Nom du cluster : Nom du cluster qui exécute ce rapport. Si le nom du cluster n'est pas défini, Vijeo Citect estime que le rapport est exécuté sur tous les clusters définis.

Heure : Heure de synchronisation du rapport exprimée en hh:mm:ss (heures:minutes:secondes). Si l'heure n'est pas spécifiée, le rapport est synchronisé à 0:00:00 (minuit).

Période : Période (fréquence) de production du rapport, en hh:mm:ss (heures:minutes:secondes).

Déclencheur : Toute Expression Cicode (ou tag de variable) peut être utilisée pour déclencher un rapport. Le rapport n'est généré que si le résultat de l'expression (figurant dans ce champ) est VRAI et si les champs Heure et Période sont vides. Il n'est généré à nouveau que si le résultat de l'expression devient VRAI après avoir été FAUX.

Fichier de format de rapport : Nom du fichier de format de rapport. L'extension de fichier par défaut est .RPT. Le compilateur Vijeo Citect suppose que le rapport utilise le format ASCII, à moins que l'extension RTF ne soit utilisée.

Périphérique de sortie : Périphérique auquel est envoyé le rapport (défini dans la section **9.1.1**)

9.1.3. Edition du fichier du rapport

C'est peut être l'étape la plus intéressante puisque elle permet de définir exactement le contenu du rapport. En cliquant sur **Edition** dans la fenêtre de configuration des propriétés u rapport, le rapport s'ouvre (à l'aide de *NotePad* par exemple si le rapport a le format .RTF) et on peut alors éditer son contenu, le rapport peut contenir un texte fixe (statique) ou des expressions Cicode qui s'exécute lors de l'appel du rapport. Les expressions et les variables Cicode doivent être placées entre ***accolades***.

On va illustrer l'édition du fichier d'un rapport à l'aide d'un exemple d'application

Exemple d'application :

On dispose d'un tableau général basse tension (TGBT) alimentant les différents système électrique d premier étage d'un bâtiment, cette TGBT est menue de PM 510 (Power Meter) permettant la mesure de la consommation électrique de chacun des systèmes de l'étage (HVAC, éclairage, appareils branchés sur les prises électriques et autres), on veut maintenant éditer un rapport pour donner la consommation énergétique journalière du système d'éclairage et celle du système HVAC (ou CVC pour Chauffage, Ventilation et Climatisation)

On se contente ici d'expliquer comment obtenir les rapports et nous n'allons pas détailler le développement de cette application.

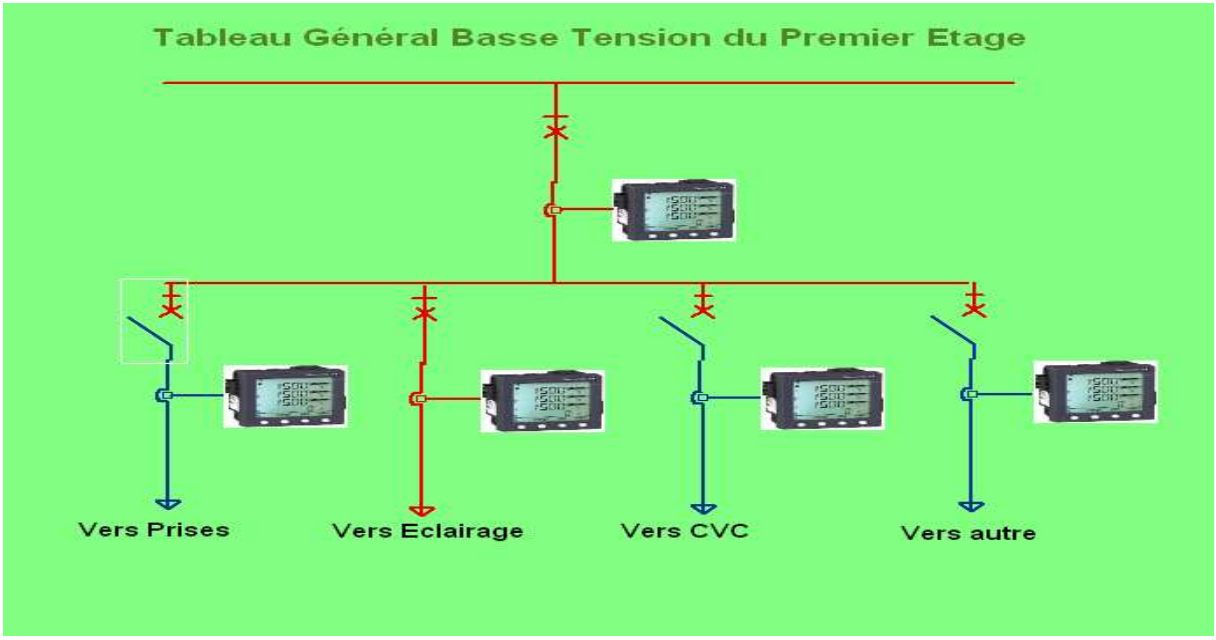


Fig. III.27. Vue de la TGBT de l'étage 1 de l'exemple d'application

On commence donc par configurer notre rapport en suivant les deux premières étapes (en veillant à fixer la période à 24 :00 :00 pour avoir un rapport quotidien).

Reste maintenant d'éditer le fichier du rapport, pour cela, comme a été déjà mentionné, dans la fenêtre de configuration du rapport, on clique sur **Edition,** le rapport s'ouvre alors dans un éditeur de texte. On écrit alors le code suivant :

```
------------------------------------------------------------------
RAPPORT DE CONSOMMATION JOURNALIERE
------------------------------------------------------------------
{Time(1) } {Date(2) }

Consommation du systeme  d'éclairage du premier etage {PuissanceActivEclairage_pm510:###.##}
KWh
Consommation du systeme HVAC du premier etage  {PuissanceActivCVC_pm510:###.##} KWh

{! Ceci est un commentaire: Affiche "Rapport de consommation achevé" à l'écran}
{CICODE}
Print("Fin du rapport")
PuissanceActivEclairage_pm510= 0; ! Réinitialise le comptage de la consommation de l'eclairage
PuissanceActivCVC_pm510= 0; ! Réinitialise le comptage de la consommation du systeme CVC
Prompt("Rapport de consommation achevé");
{END}
{FF}
```

Remarques :

- Tout ce qui est placé après le point d'exclamation '!' est un commentaire

- Le tag *''PuissanceActivEclairage_pm510''* (type Real) est relative à la PM donnant la consommation du système d'éclairage.
- Le tag *''PuissanceActivCVC_pm510''* (type Real) est relative à la PM donnant la consommation du système CVC.
- On veillera bien à ne pas oublier de définir les tags dans le logiciel UnityPro et de leurs assigner les mêmes adresses que dans Vijeo Citect.
- Les deux expressions {CICODE} et {END}délimitent un bloc de code CICODE.

Ci-dessous est donné un exemple d'exécution du rapport précédent

```
-----------------------------------
RAPPORT DE CONSOMMATION JOURNALIERE
-----------------------------------
6:00am    12/3/2010
Consommation du systeme  d'éclairage du premier etage  0020.50  KWh
Consommation du systeme HVAC du premier etage         0042.30 KWh

Fin du rapport
```

Remarque : Il est nécessaire de définir le PC comme Serveur de rapports à l'aide de l'Assistant de configuration de l'ordinateur avant de configurer les rapports.

Conclusion

Durant ce chapitre, nous avons présenté de façon succincte le logiciel Vijeo Citect ainsi que certaines de ces caractéristiques et fonctionnalités, certaines options du logiciel on été illustrés à l'aide d'un exemple d'application de façon à simplifier l'apprentissage.

Nous avons montré que grâce à ces fonctionnalités avancées, Vijeo Citect constitue une solution SCADA complète et homogène.

A noter que ce chapitre ne se prétends nullement d'être un manuel d'apprentissage du logiciel mais plus tôt un petit guide présentant quelques unes des principales fonctionnalités du logiciel et permettant une prise en main rapide par l'apprenti. Pour une utilisation plus avancée, la consultation d'un manuel et du Help du logiciel sont nécessaires.

CHAPITRE IV

Présentation de l’Application

1. Introduction et descriptif général de l'application

Au cours de ce chapitre, nous allons présenter l'application mise en œuvre dans le cadre de notre Projet de Fin d'étude. En effet, en utilisant les concepts et les outils décrits lors des chapitres précédents (outils tel que le logiciel Unity Pro ou encore le logiciel SCADA Vijeo Citect), nous allons créer notre propre application SCADA permettant le contrôle et la supervision d'une tour de bureaux, cette application va permettre le contrôle et la gestion de divers systèmes à savoir :

1. La gestion technique d'électricité, (la gestion de l'ensemble de l'installation électrique alimentant notre bâtiment).
2. La gestion du système d'éclairage (intérieur et extérieur).
3. La Gestion du système HVAC (Heating, Ventilating and Air Conditioning) à travers le contrôle des **CTA** (**Centrale de Traitement d'Air**).

L'application développée sera dotée de toutes les fonctionnalités d'une application SCADA complète, elle permettra entre autres d'effectuer les tâches suivantes :

1. Superviser toute l'installation et déterminer l'état de chacun des ses sous systèmes : état de tous les disjoncteurs de protection de notre installation électrique, état de l'éclairage dans chacune des zones de la tour, état des différentes CTA permettant le contrôle de la température…etc.
2. Commander les différents sous systèmes de notre installation : ouverture des disjoncteurs des différents systèmes électrique, allumage ou extinction de l'éclairage directement du superviseur (on reviendra sur tous les modes de gestion de l'éclairage plus loin), modification des différentes consignes tel que la consigne de température d'air repris de la CTA…etc.
3. Signaler à l'opérateur les différentes alarmes qui surviennent lors du fonctionnement du système : alarme signalant la coupure du réseau électrique, alarme signalant le déclenchement d'un disjoncteur donné...etc.

1.1. Description de la tour

Notre tour dispose d'un sous sol, d'un rez-de-chaussée et de six niveaux ainsi que de zones extérieures dont on peut commander l'éclairage :

- Le sous sol comporte les différents systèmes électrique (les TGBT : Tableau Général Basse Tension, Groupe Electrogène, Onduleur…etc.), il est subdivisé en 10 zones d'éclairage (zone 1A jusqu'à 6A et zone 1B à 4B)
- Le rez de chaussée est subdivisé à son tour en 14 zones d'éclairage (de 1A à 14A)
- Chacun des six niveaux est subdivisé en 17 zones d'éclairage (de 1A à 10A et de 1B à 7B), un niveau comporte donc deux tableau divisionnaires TDA et TDB relatifs aux zones A et B, ces tableaux comporte les télérupteurs de commande d'éclairage ainsi

que les disjoncteurs de protection. Les niveaux disposent aussi des CTA permettant la climatisation.

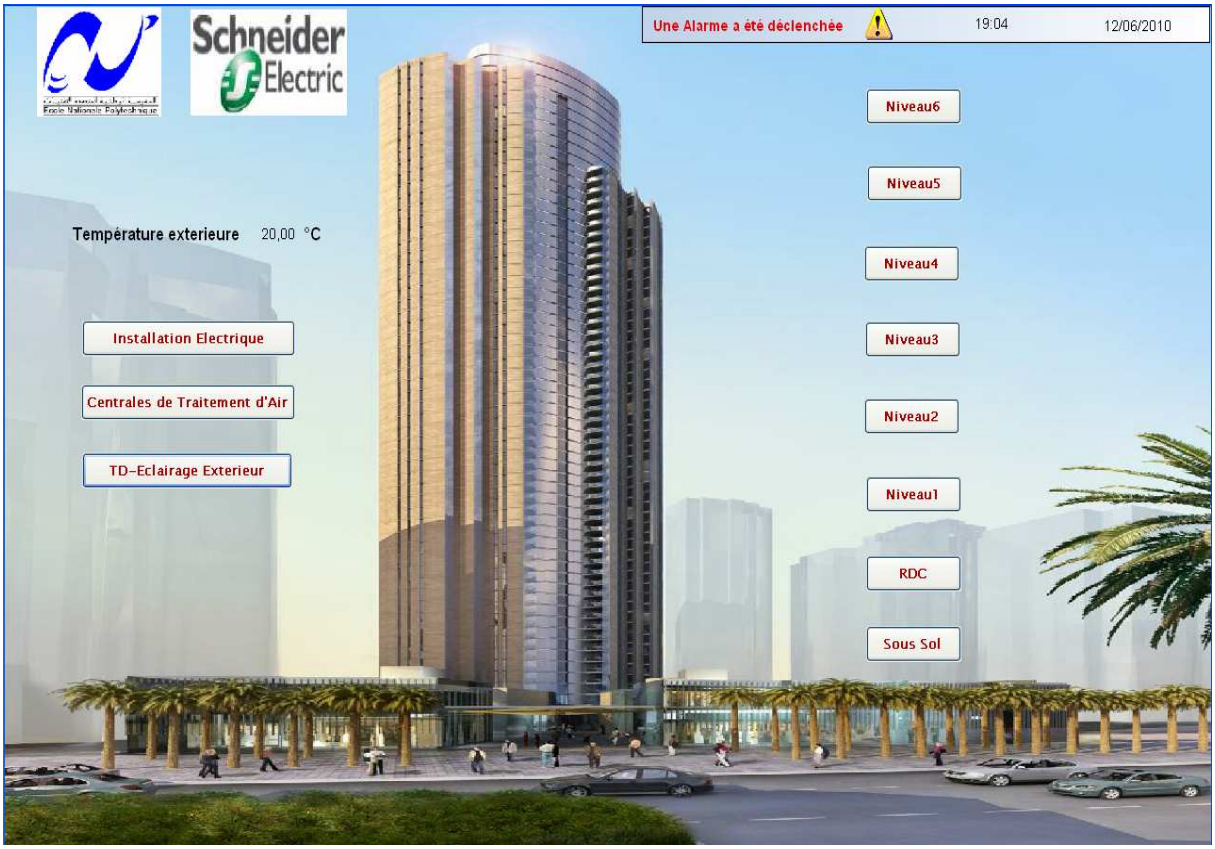


Fig. IV.1. Page d'Accueil du projet

La figure ci-dessus illustre la page d'accueil de notre projet, elle comporte des raccourcis vers tous les niveaux de notre tour (le sous sol, le rez-de-chaussée ainsi que les six niveaux), elle compote aussi des raccourcis vers les principaux systèmes composant notre tour : l'installation électrique, les centrales de traitement d'air (CTA) ainsi que le tableau divisionnaire de l'éclairage extérieur. Les boutons de raccourcis des niveaux permettent d'accéder au système de commande de l'éclairage du niveau correspondant.

2. Description de l'installation électrique

La **figure IV.2** donne le synoptique général de l'installation électrique de notre bâtiment, on y voit les éléments suivants : un poste de livraison, deux TGBT, des Tableaux Divisionnaires (TD) pour l'éclairage, deux groupes électrogènes de secours avec un TGBT Groupe, un onduleur avec son TGCO (Tableau Général du Courant Ondulé). En cliquant sur un élément dans le synoptique, on va directement vers la page correspondante.

Dans ce qui suit, on donnera un descriptif général des éléments de notre installation électrique.

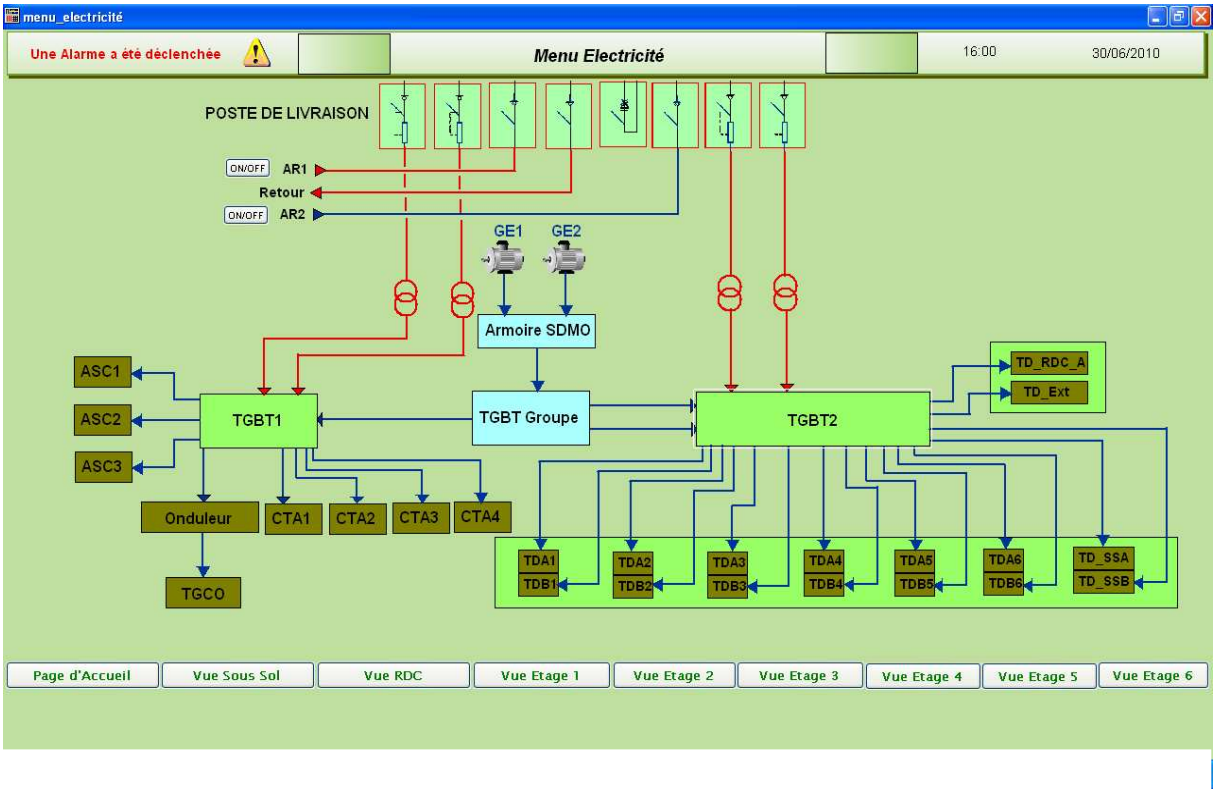


Fig. IV.2. Synoptique général de l'installation électrique

2.1. Le Poste de livraison

Alimenté par deux arrivées de 10KV, dont une seule est fonctionnelle à la fois, la deuxième arrivée est considérée comme une arrivée de secours en cas de coupure électrique dans la première.

Le schéma du poste de livraison est illustré sur la figure IV.3.

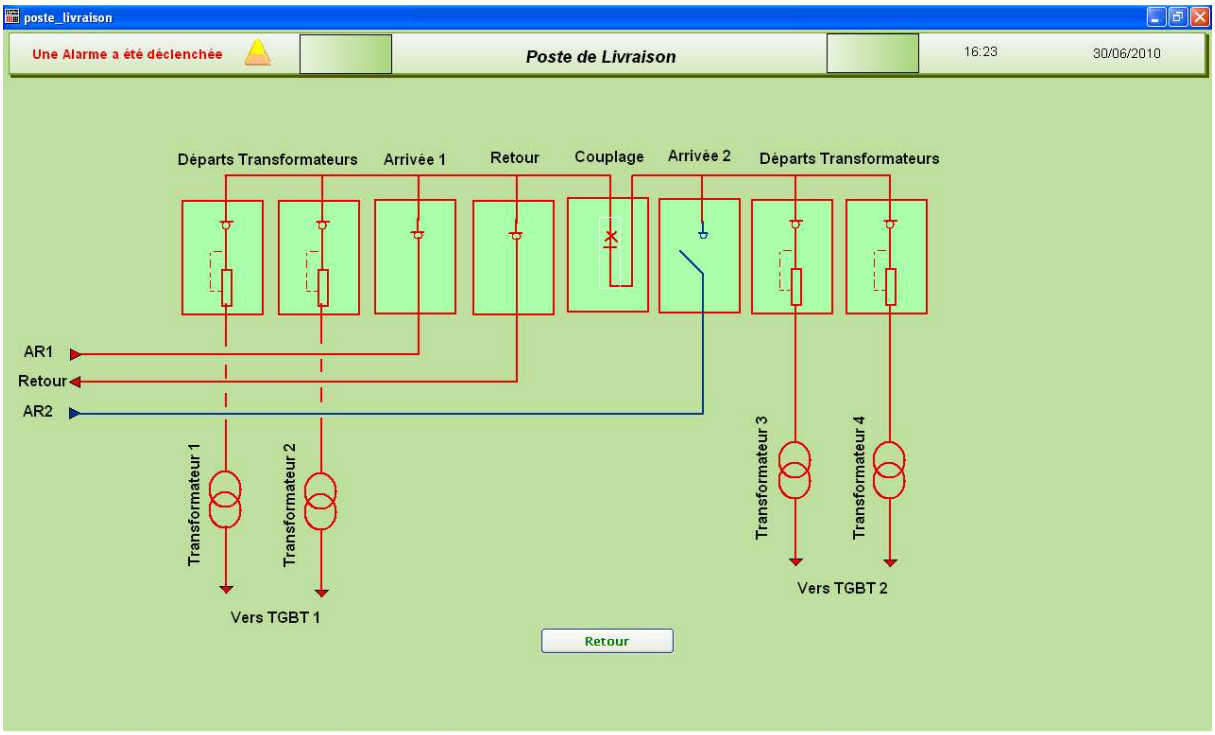


Fig. IV.3. Schéma électrique du poste de livraison

Le poste de livraison est constitué de huit cellules remplissant divers fonctions, on distingue :

- Deux cellules de raccordement au réseau contenant chacune un interrupteur sectionneur 10KV **(**sur la figure schématisé par Arrivée 1 et Arrivée 2), ces deux interrupteurs ne sont jamais fermés simultanément.
- Une cellule de départ 'Retour' permettant d'alimenter un autre client.
- Une cellule de couplage contenant un disjoncteur de couplage, ce disjoncteur est toujours fermé sauf lorsqu'on veut isoler un TGBT (pour des raisons de maintenance par exemple).
- Deux cellules de départ vers transformateurs (moyenne puissance) contenant des interrupteurs fusibles.
- Deux cellules de départ vers transformateurs (grande puissance) contenant des interrupteurs fusibles.

Il est à signaler que mis à part l'interrupteur de couplage, aucun des six autres interrupteurs du poste de livraison n'est commandable par l'opérateur à partir du superviseur.
Le fonctionnement du système est décrit comme suit :

- Par défaut, l'interrupteur de la cellule Arrivée 1 est fermé, celui de l'Arrivée 2 est ouvert et le coupleur est fermé, les deux TGBT sont ainsi alimentés par cette arrivée 1.
- Dès qu'il y'a coupure sur l'arrivée 1, l'interrupteur de cette arrivée s'ouvre, celui de l'Arrivée 2 se ferme et l'installation est ainsi alimentée par la deuxième arrivée.
- Lorsque le réseau sera rétabli sur l'arrivée 1, l'interrupteur de l'arrivée 2 s'ouvre, et on rebascule sur l'arrivée 1.

Le programme Unity Pro permettant la gestion du poste de livraison est donné en **Annexe E**



Fig. IV.4 Exemple de la disposition des différentes cellules dans un poste de livraison

2.2. Les transformateurs

Comme illustré sur la **figure IV.3**, notre installation est dotée de 4 transformateurs assurant la liaison entre le poste de livraison et les TGBT.

On dispose de deux transformateurs de 1250KVA assurant la liaison entre le poste de livraison et le TGBT 1 et de deux autres transformateurs de 630KVA assurant la liaison avec le TGBT 2

Remarque : Dans les grands immeubles, il est interdit d'installer des transformateurs contenant plus de 25 litres de diélectrique liquide ; ce qui rend obligatoire utilisation des transformateurs sec enrobé dans cette installation.
Dans un transformateur sec, Le circuit magnétique est isolé (ou enrobé) dans une matière isolante sèche à plusieurs composants. Ce type de transformateur a l'avantage de ne présenter aucun risque de fuite ou de pollution, en revanche il nécessite des précautions de mise en œuvre et de maintenance (local ventilé, dépoussiérage…). Le refroidissement est assuré grâce à l'air ambiant sans liquide intermédiaire.



Fig. IV.5. Exemple de la disposition des transformateurs dans une tour

2.3. Le TGBT 1

Le premier Tableau Général Basse Tension (TGBT) est utilisé pour l'alimentation des installations énergivores, à savoir les CTA (Centrales de Traitement d'Air), le GEG (Groupe d'Eau Glacée), les ascenseurs, l'onduleur et la pompe de relevage. C'est pourquoi ce tableau est alimenté par les transformateurs ayant une plus grande puissance (les deux transformateurs de 1250KVA)

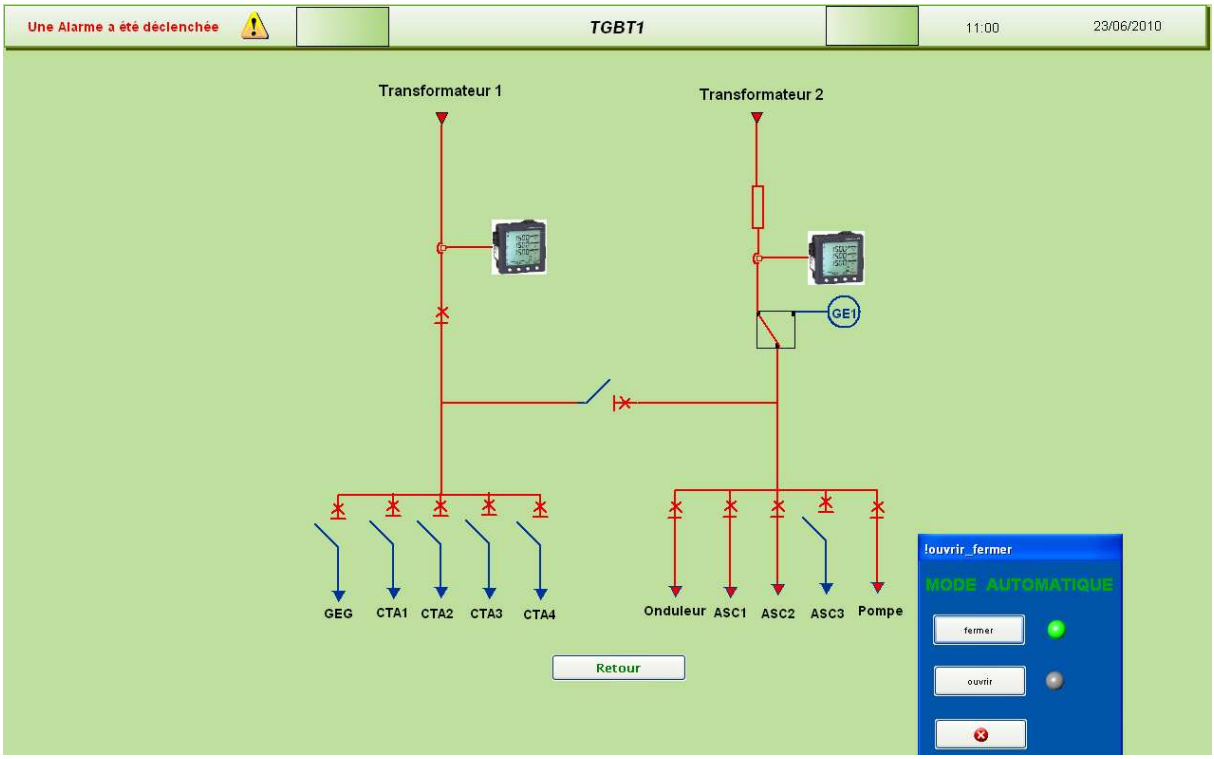


Fig. IV.6. Schéma électrique du tableau TGBT 1

La **figure IV.6**. illustre le schéma électrique du TGBT 1 où on peut visualiser les deux arrivées. Un inverseur de source est prévu sur l'arrivée venant du transformateur 2 alimentant l'onduleur, les ascenseurs et la pompe de relevage. Cet inverseur de source permet le basculement vers le générateur en cas de coupure de réseau, ainsi, on voit que les installations les plus énergivores (les CTA et le GEG) ne sont pas secourus afin de ne pas surcharger le générateur (de plus, ces installations ne sont pas d'une importance capitale comme le sont les ascenseurs, l'onduleur et la pompe de relevage, et le délestage effectué ne sera donc pas préjudiciable pour les occupants de la tour).

Les éléments constituant le TGBT 1 (schématisés sur la **figure IV.6**) sont les suivants :

- Un disjoncteur sectionneur de tête, du coté du transformateur 1 ; c'est le **Masterpact 2500**. Les détails concernant ce disjoncteur sont donnés en **Annexe D**.
- L'inverseur de source décrit précédemment et permettant de secourir le coté du transformateur 2. C'est un Masterpact NW 2500 (description donnée en **Annexe D**)

- Un disjoncteur de couplage (un Masterpact 2500) permettant de coupler les deux cotés en cas d'absence d'alimentation sur l'une des deux arrivée pour assurer la continuité du fonctionnement (sauf lorsqu'il y a coupure du réseau et que le coté du transformateur 2 est secouru par le générateur). Ce disjoncteur n'est donc pas actionnable directement par l'opérateur car une fausse manœuvre (fermeture du disjoncteur lorsque les deux arrivées sont alimentés) causera des dégâts néfastes pour l'installation. Le programme permettant la commande de ce disjoncteur par l'automate est donné en **Annexe E**.
- Un disjoncteur de départ pour l'onduleur, à savoir le Compact NS 1000.
- Des disjoncteurs de départ vers les quatre CTA et le GEG, ce sont des Compact NSX 400 (voir **Annexe D** pour une description sur la gamme Compact NSX)
- Des disjoncteurs de départ vers les trois ascenseurs et vers la pompe de relevage (Compact NSX 250A)

Remarque :

1. La couleur d'un câble a été programmée de façon à ce qu'elle devienne rouge lorsque ce câble est sous tension électrique et bleu dans le cas inverse. L'état d'un câble est obtenu en exploitant les informations relatives aux états des disjoncteurs en amont (l'état d'un disjoncteur est obtenu grâce à ses contacts auxiliaires) ainsi que des états des câbles les traversant.
2. On a programmé des supers génies pour la commande des disjoncteurs, en effet, en cliquant sur un disjoncteur, une fenêtre apparait et permet d'avoir le mode du disjoncteur (local ou automatique) et permet aussi de changer son état (l'ouvrir ou le fermer s'il est en mode automatique).



Fig. IV.7. Exemple de disposition d'armoire dans un TGBT

2.4. Le TGBT 2

Le TGBT 2 est utilisé pour l'alimentation des tableaux divisionnaires (TD) de l'éclairage. Ce TGBT est alimenté donc par les deux transformateurs de 630 KVA, il est doté de deux

inverseurs de source permettant de basculer sur les groupes électrogènes lorsqu'il y a coupure du réseau. La liaison avec le groupe électrogène se fait à travers le Tableau Général du Groupe Electrogène (TGGE) qu'on verra par la suite.

La **figure IV.8** illustre le schéma électrique du TGBT 2, on peut identifier les éléments suivants :

- Un disjoncteur de couplage (le Compact NS 1000), ce disjoncteur permet de coupler les deux cotés en cas d'absence d'alimentation sur l'une des deux arrivée pour assurer la continuité du fonctionnement, il est commandé uniquement par l'automate (le programme est donnée en **Annexe E**).
- Des disjoncteurs de départ vers les tableaux divisionnaire de l'éclairage des différents niveaux. Ces 16 disjoncteurs sont commandables par l'opérateur à partir de la station de supervision. La modèle de ces disjoncteurs est le Compact NSX 160 (description donnée en **Annexe D**).

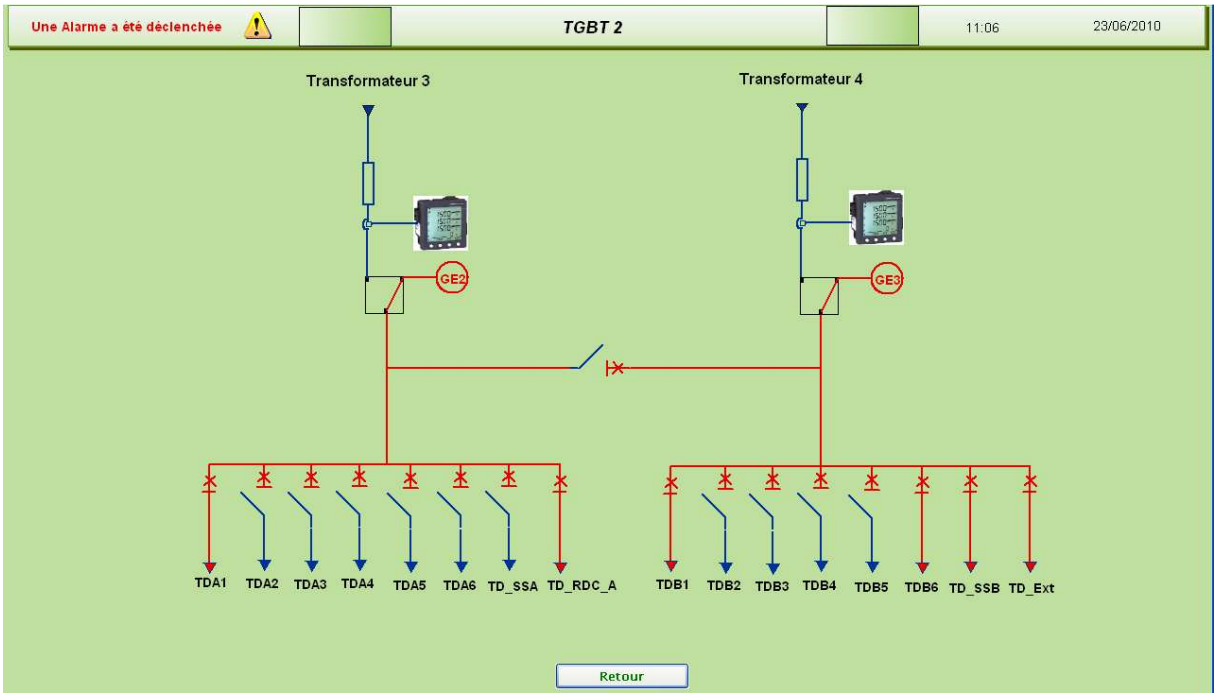


Fig. IV.8. Schéma électrique du TGBT 2

2.5. Le Tableau Général du Courant Ondulé

Notre installation électrique dispose aussi d'un Tableau Général du Courant Ondulé (TGCO). Ce tableau est alimenté par un onduleur et comporte un disjoncteur de tête pour la protection (le Compact NS 1000) et des disjoncteurs de départs permettant d'alimenter les Coffrets du Courant ondulé des zones A et B relatifs aux six niveaux (des Compact NSX 160), ainsi, on dispose en tout de 13 disjoncteurs dans le TGCO.

Remarque : Le courant ondulé est obtenu à partir de l'onduleur, c'est donc un courant plus lisse et comporte moins d'harmoniques relativement à celui obtenu du générateur ou de

l'arrivée réseau, si bien qu'il est généralement utilisé pour l'alimentation du matériels informatique.

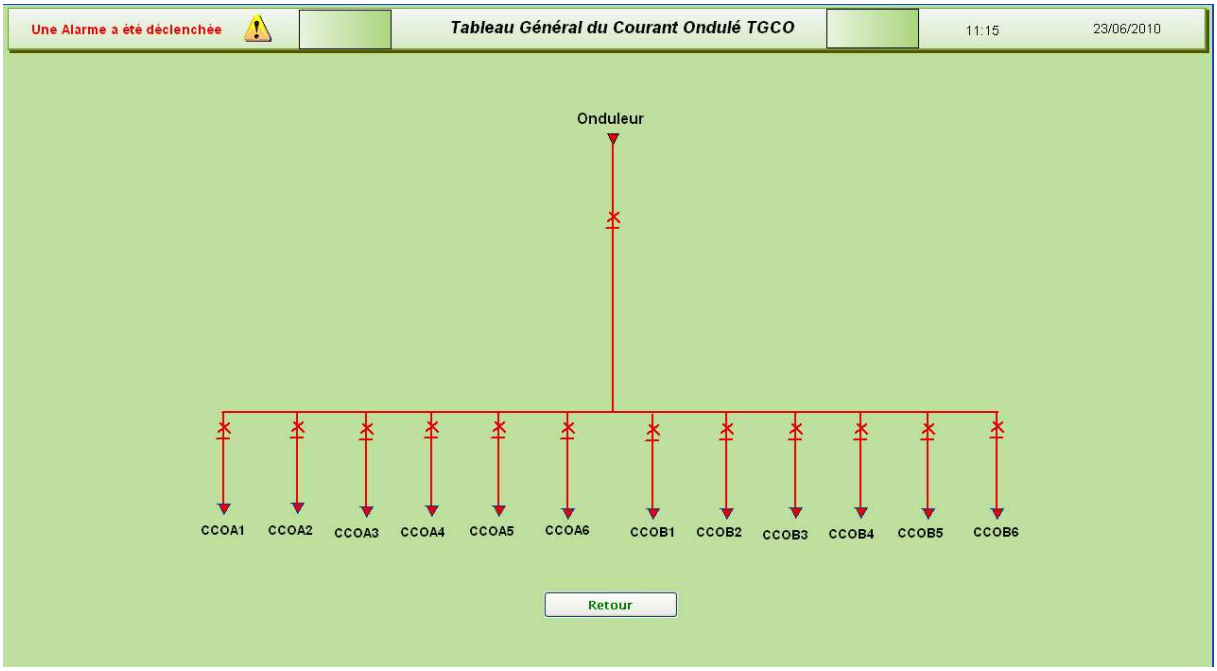


Fig. IV.9. Tableau Général du Courant Ondulé

2.6. Le Tableau divisionnaire d'éclairage

On a vu auparavant que le TGBT 2 comporte des disjoncteurs de départ pour l'alimentation de neuf Tableaux Divisionnaires (TD) d'éclairage : les six TD correspondant aux six niveaux, ainsi que celui du rez-de-chaussée, du sous sol, et de l'éclairage extérieur. La **figure IV.10** illustre le schéma électrique d'un tableau divisionnaire d'éclairage (ici le TD du 1er niveau). Il comporte deux disjoncteurs de tête pour chacune des deux zones A et B (le Compact NSX 160) ainsi que des disjoncteurs de départs pour la protection (des disjoncteurs de la série C60N ayant comme courant assigné 16A et comme courant de déclenchement 10 kA, les détails sont donnée en **Annexe D**)

Seuls les deux disjoncteurs de tête sont commandables, les disjoncteurs de départ ne sont prévus que pour la protection, en effet, la commande de l'éclairage de chaque zone se fait par action sur un télérupteur (non schématisé sur cette figure, une description sur les télérupteurs sera donnée en **Annexe D**) ; la page permettant la commande d'éclairage de chaque zone sera illustrée plus loin.

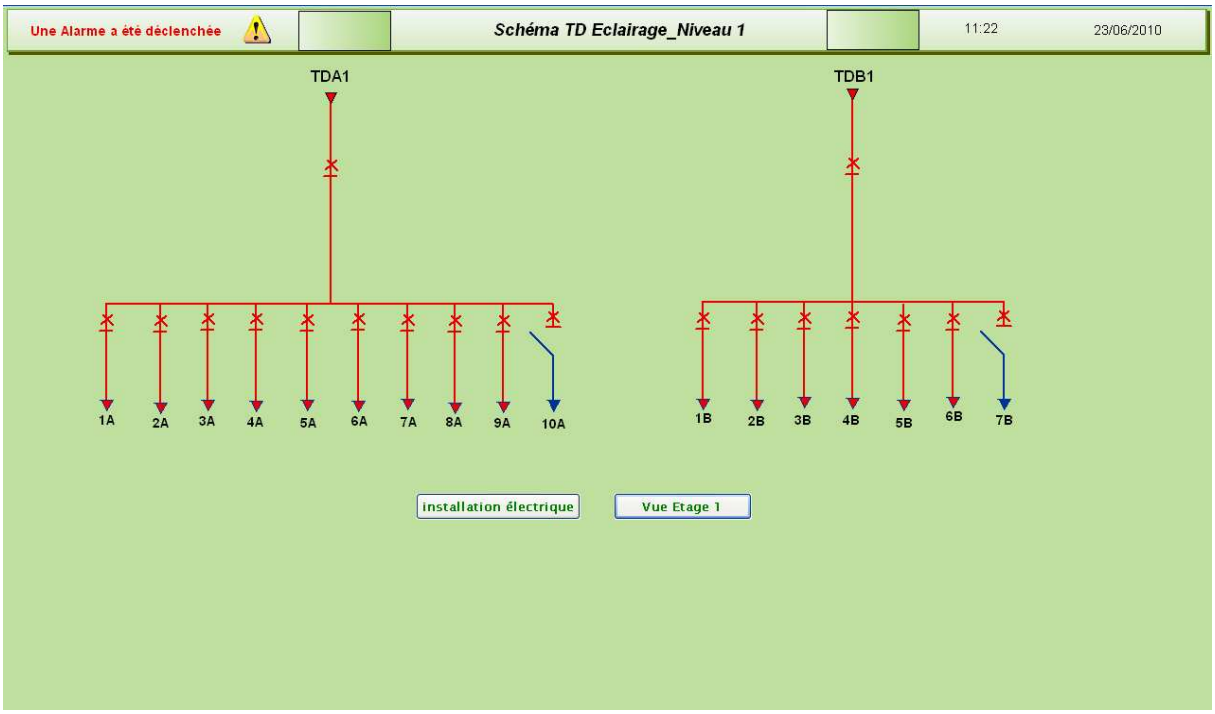


Fig. IV.10. Le tableau divisionnaire pour l'éclairage du premier étage

2.7. Le Tableau Général du Groupe Electrogène TGGE

Notre installation dispose de deux groupes électrogènes à utiliser en cas de coupure du réseau sur les deux arrivées, la puissance des deux groupes étant couplée au niveau de l'armoire SDMO, la sortie de cette armoire alimente le tableau TGGE qui dispose d'un disjoncteur de tête et de trois disjoncteurs de départ vers les TGBT. Tous les disjoncteurs de ce tableau sont commandables par l'opérateur.

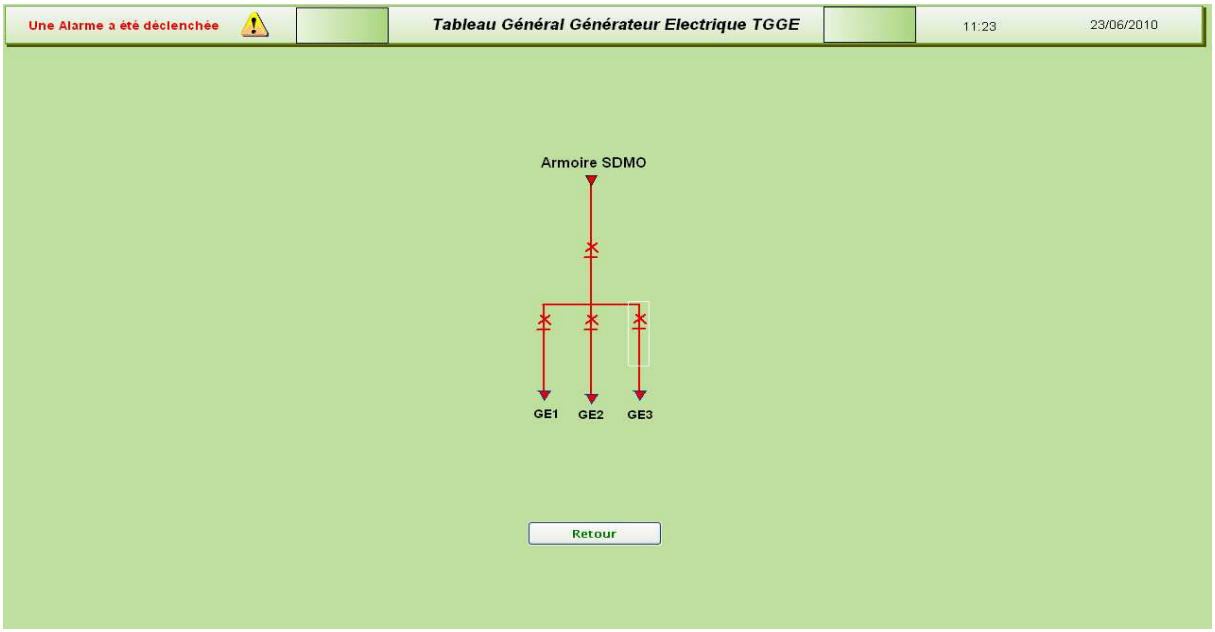


Fig. IV.11. Schéma électrique du Tableau Général du Groupe Electrogène

2.8. Les alarmes

Pour tous projets de GTB, la configuration et l'affichage des alarmes est une étape primordiale qui permet à l'opérateur de savoir à tout moment si une panne ou une anomalie est survenue dans le système, ce qui permettra de prendre les mesures nécessaires dans les plus brefs délais afin de palier au problème.

Aussi, cette étape importante n'a pas été omise dans notre projet et plusieurs alarmes ont été prévues et configurées En effet dès qu'une alarme survient, un symbole d'alarme (le triangle en haut à gauche de l'écran) commence à clignoter et une écriture rouge apparait indiquant qu'une alarme a été déclenchée.

En cliquant sur le symbole d'alarme, on peut aller sur la page d'alarme (**figure IV.12**) pour visualiser toutes les alarmes, cette page donne des informations sur la date et l'heure du déclenchement de l'alarme, le nom et la description de cette dernière, ainsi que son état (lorsque la condition déclenchant l'alarme n'est plus valide, l'état est alors faux). Cette page permet à l'opérateur d'acquitter une alarme en cliquant dessus.

En utilisant le bouton de navigation sur la page d'alarme (indiqué par la flèche bleue), on peut aussi se rendre vers une autre page donnant l'historique et le sommaire de toutes les alarmes déclenchées avec plus de détails (tel que la catégorie de l'alarme, la date et l'heure de l'acquittement, la durée, l'état…etc.)

Remarque :

- On a configuré notre application pour donner un fichier d'historique de toutes les alarmes déclenchées, ce fichier est sous format *DBF* (Data Base File) et peut être trouvé dans le répertoire de sauvegarde du projet.
- La coupure du réseau électrique, le déclenchement des disjoncteurs de TGBT, du groupe électrogène, ceux du TGCO,…etc. tous ces événements ont été prévu comme des alarmes pour que l'opérateur soit averti dès coupure de l'alimentation électrique d'un endroit donné.
- D'autres alarmes relatives aux systèmes d'éclairage et de climatisation vont être discutées plus loin.

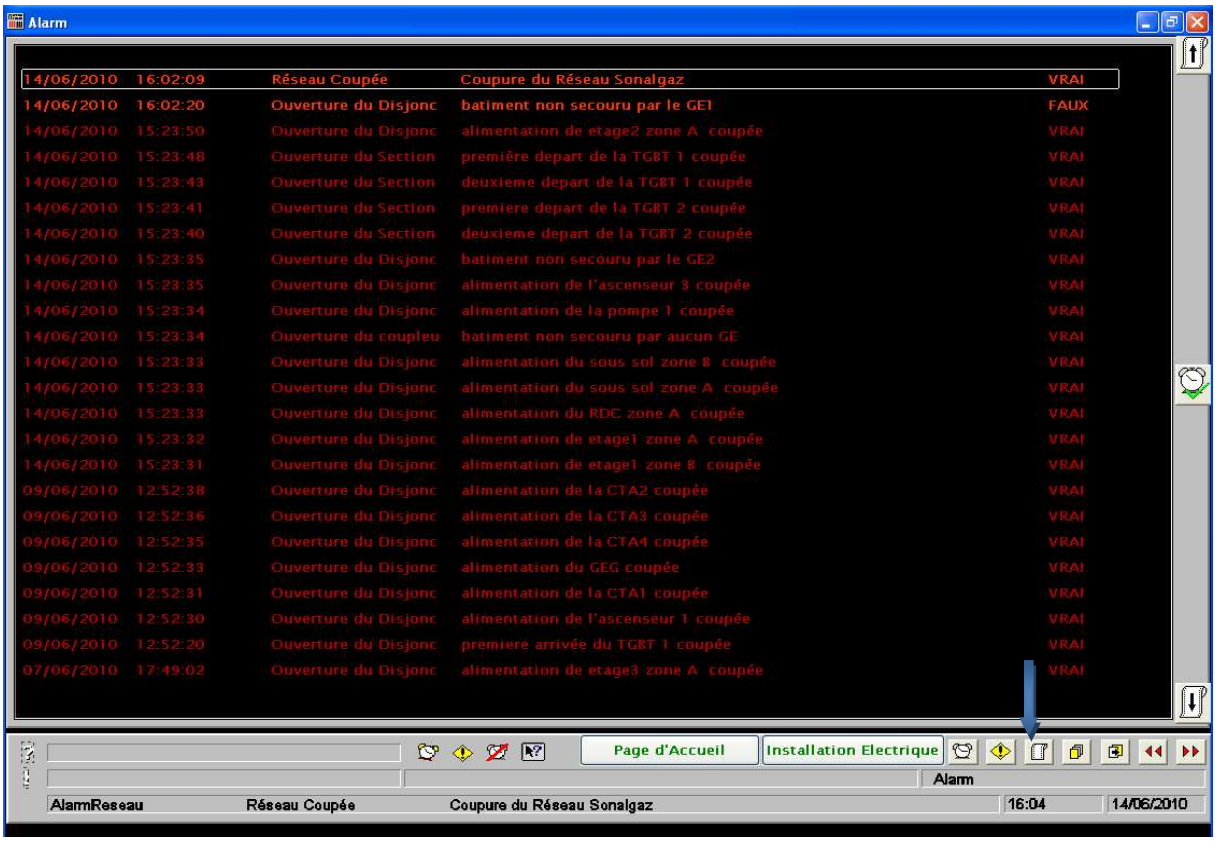


Fig. IV.12. Page d'alarme

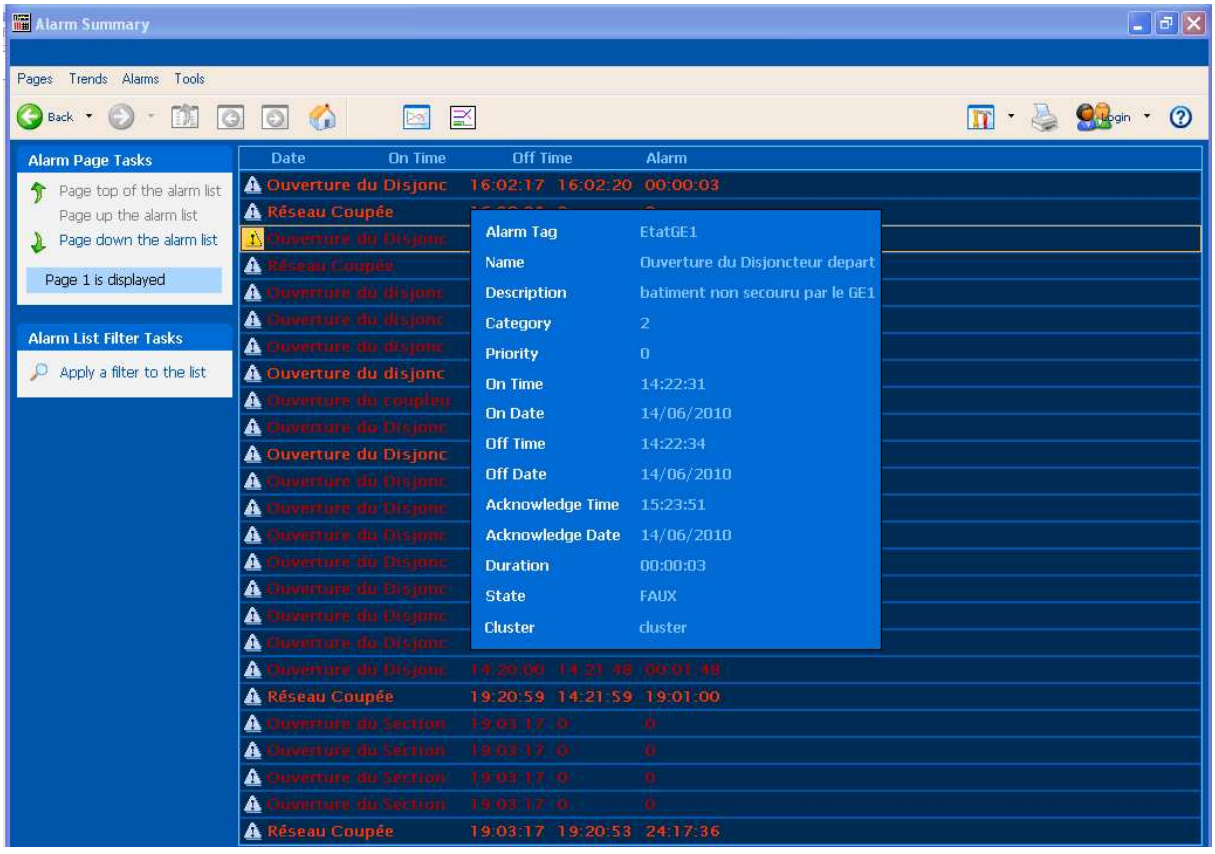


Fig. IV.13. Page sommaire des alarmes

3. description du système d'éclairage

Pour la commande du système d'éclairage de notre tour, les modes de gestion suivants ont été prévus :

1. **Mode Manuel** : où l'opérateur peut allumer ou éteindre l'éclairage dans une zone en appuyant sur un bouton à partir de la station de supervision.
2. **Mode de gestion horaire (Automatique)** : Ce mode permet à l'opérateur de saisir l'heure d'allumage et d'extinction d'une zone à partir de la station de supervision.
3. **Mode de gestion en fonction de la présence et de l'apport en éclairage naturel** : Ce mode, implémenté uniquement dans les endroits enregistrant des périodes d'absence assez prolongée (bureau clos de réunion par exemple) permet de gérer l'éclairage tout en tenant compte de la présence des personnes ainsi que de l'éclairage naturel, ce qui permet d'avoir une gestion énergétiquement plus efficace.
4. **Mode crépusculaire** : ce mode, prévu exclusivement pour l'éclairage extérieur, permet d'allumer ou d'éteindre la lumière extérieure selon l'état de la lumière du jour. Pour cela on utilise un capteur crépusculaire ayant une sortie logique, cette sortie est mise à 1 dès que le niveau d'éclairage descend au dessous d'un seuil réglable donnée.

Dans ce qui suit, on donnera un descriptif des systèmes d'éclairage intérieur et extérieur respectivement et où l'opérateur peut choisir entre l'un des modes de gestion décrits précédemment.

3.1. Description du système d'éclairage intérieur

Pour la gestion du mode d'éclairage intérieur, on dispose d'un tableau permettant le choix du mode de gestion pour chacune des zones d'un étage donnée (**figure IV.14**.)

Pour le mode manuel, l'opérateur dispose d'un bouton permettant l'allumage ou l'extinction d'une zone donnée. Un voyant permet de déterminé l'état de l'éclairage de la zone.

Si l'opérateur choisit le mode automatique (ou mode de gestion horaire), le premier mode sera automatiquement désactivé, le bouton d'allumage et d'extinction sera à son tour grisé (désactivé) et l'opérateur pourra alors se rendre vers la page de Gestion Horaire pour introduire les heures d'allumage et d'extinction d'une zone donnée.

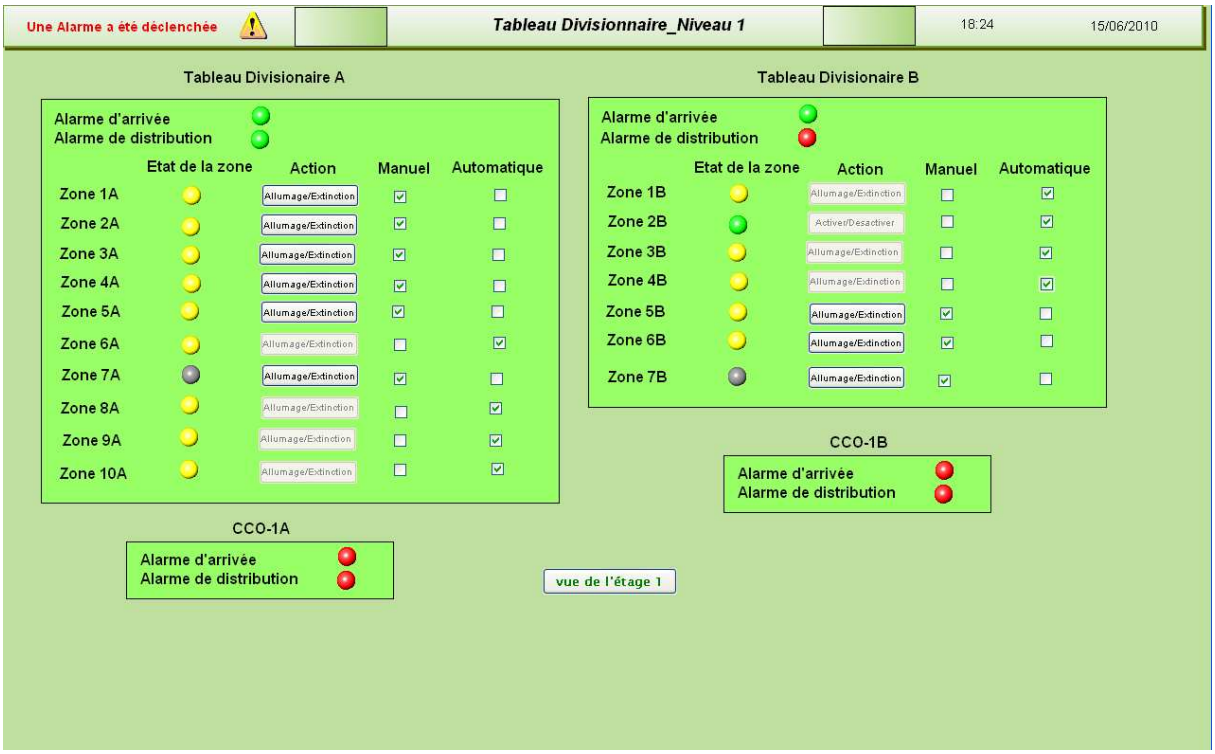


Fig. IV.14 Page de commande d'éclairage du premier étage

Remarque :

- En plus du symbole d'alarme disposé en haut à gauche de l'écran, la page dispose aussi de deux indicateurs d'alarmes ; **Alarme d'arrivée** et **alarme de distribution**. La première indique que le disjoncteur de tête du tableau divisionnaire est ouvert, tandis que l'**Alarme d'arrivée** indique que l'un des disjoncteurs de protection est ouvert, l'opérateur pourra alors vérifier quel disjoncteur a été déclenché en se rendant vers la page du Tableau divisionnaire correspondant
- Il est évident de noter que l'éclairage d'une zone sera désactivé dès que cette zone n'est plus alimentée en électricité.

La figure ci-dessous illustre la page permettant la gestion horaire relative au premier étage. On voit que pour chacune des zones, on peut saisir l'heure d'allumage et celui d'extinction, un bouton doit être appuyé ensuite pour confirmer les modifications des heures.



Fig. IV.15. Page de Gestion horaire du premier étage

Remarque :

- Seules les zones configurées par le mode de gestion horaire (celles qui ne sont pas en mode manuel) seront affectées par les modifications des heures d'allumage et d'extinction.
- L'heure d'allumage et d'extinction d'une zone donnée étant variable et saisies par l'opérateur, on ne peut donc utiliser l'option *Evènements* de Vijeo Citect pour programmer l'allumage et l'extinction d'une zone, en effet ceux-ci ont été programmé en Cicode (langage évolué de Vijeo Citect et rassemblant au C) le programme Cicode de gestion horaire sera donnée en **Annexe E**.

La figure ci-dessous illustre la vue de l'étage 1, on voit que les zones allumées deviennent éclairées en jaune, les zones non éclairés sont sombres (grises), la page comporte aussi des boutons de navigation qui permettent de se rendre vers la page de commande d'éclairage (choix du mode et commande), la page de gestion horaire, le tableau divisionnaire ainsi que la page d'accueil et celle de l'installation électrique.

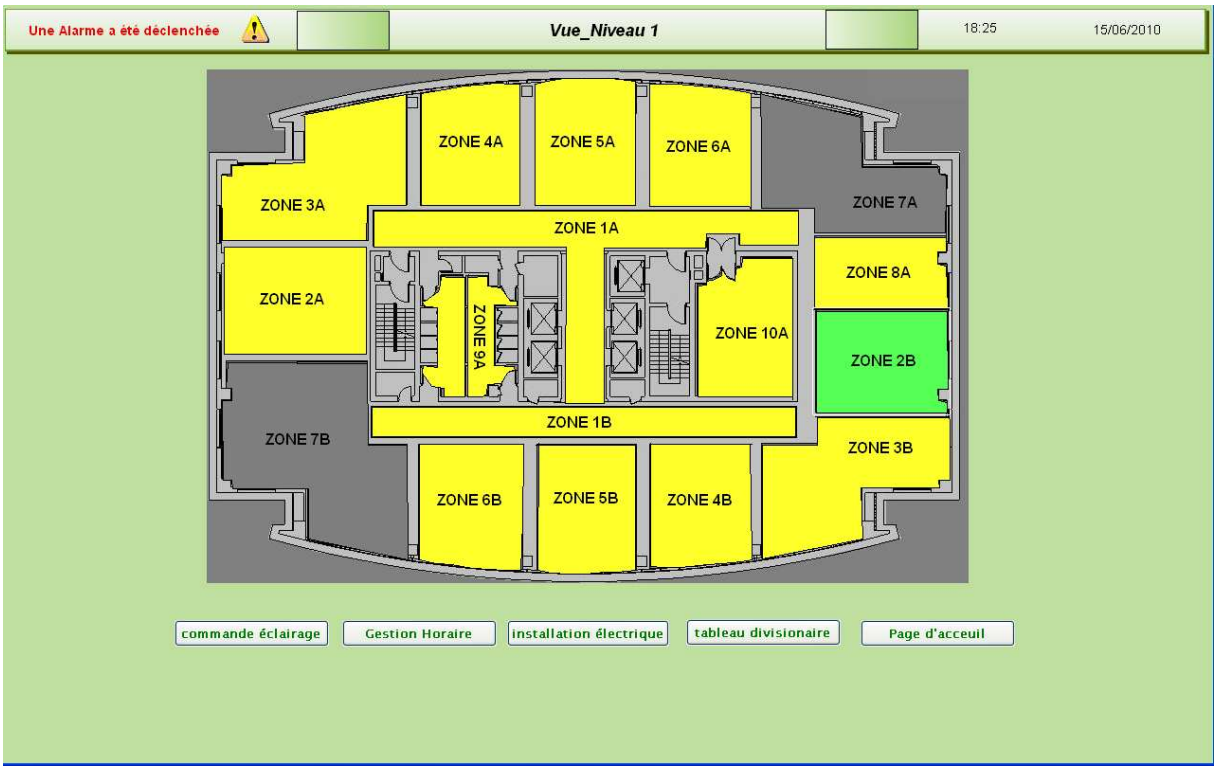


Fig.IV.16. Vue de l'étage 1

3.1.1 Gestion de l'éclairage en fonction de la présence et de l'apport en éclairage naturel

Les deux modes vus précédemment (mode manuel et mode de gestion horaire) ne sont appliqué généralement que dans les espaces ouverts (*Open Space*) qui enregistrent un mouvement continu du personnel et où il sera gênant et non efficace de prévoir un mode de gestion en fonction de la présence des occupants en raison des allumages-extinctions fréquents.

Cependant, pour les espaces fermés qui enregistrent en général des périodes de non occupation assez longues (bureau, salle de réunion…etc.), adopter une politique de gestion horaire de l'éclairage sera irrationnel et ne serai pas en concordance avec l'efficacité énergétique (un des objectifs de la GTB), et il sera donc judicieux et économiquement rentable de prévoir des capteurs de mouvement afin d'éteindre l'éclairage lorsque le local n'est pas occupé.

Tenant compte de ce qui a été dit et afin de réduire la facture de la consommation de l'éclairage, nous avons implémenté dans notre application (au niveau de la zone 2B) une gestion de l'éclairage en fonction de la présence des occupants et de l'éclairage naturel, le cahier de charge est donné ci-dessous :

Cahier de charge :

Nous disposons d'un bureau muni de trois rangées de luminaires pour l'éclairage et de trois détecteurs de mouvement (chacun associé à une rangée) servant comme détecteur de présence. On veut que l'éclairage soit commandé par le détecteur de mouvement afin de

l'éteindre lorsqu'aucune présence n'est enregistrée pour rendre ainsi la consommation d'énergie plus rationnelle.

- Une rangée de luminaire est allumée si le détecteur de mouvement associé détecte une présence, sinon elle est éteinte au bout de 2 mn après la dernière détection de mouvement.
- De plus, le bureau dispose aussi de trois boutons poussoirs permettant l'allumage d'une rangée de luminaire même si le capteur de mouvement associé est désactivé (aucune présence n'est détectée), ces boutons peuvent être utiles lorsqu'un occupant souhaite allumer d'autres rangées que celle sous laquelle il travaille dans le but d'augmenter la luminosité.
- L'allumage d'une rangée par un bouton poussoir sera activé pour une durée déterminée (20 mn), au bout de cette durée, les luminaires de la rangée vont s'éteindre dans le cas où le bureau est vacant.
- La lumière des luminaires les plus proches de la fenêtre est modulée (régulée) en fonction de la lumière du jour et ceci en utilisant un capteur de lumière et un régulateur PI.

Le grafcet matérialisant ce cahier de charge est illustré ci-dessous

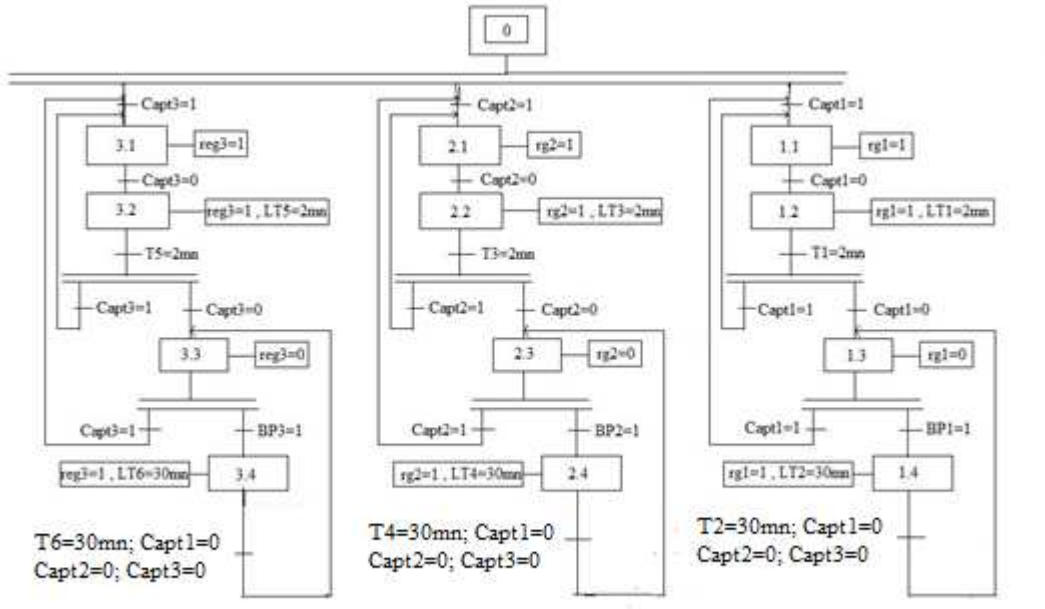


Fig. IV.17. Grafcet du cahier de charge de l'éclairage de la zone 2B

La **figure IV.18** donne une schématisation de notre bureau, on peut visualiser les différents luminaires, les trois boutons poussoirs, le capteur de lumière ainsi que les capteurs de mouvement. On dispose aussi d'un tableau permettant de saisir les paramètres du régulateur PI ainsi que d'un autre permettant de saisir la consigne d'éclairement sur la première rangée

en lux, ce tableau permet aussi de visualiser la mesure donnée par le capteur de luminosité convertie en lux

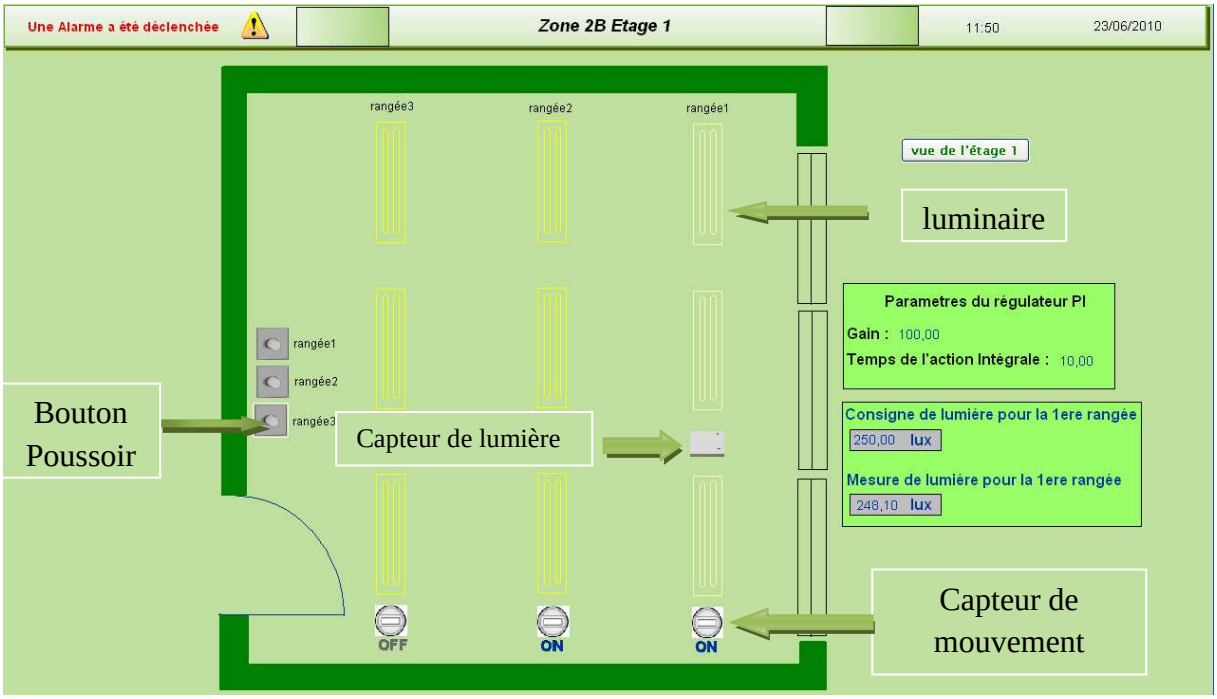


Fig. IV.18. Gestion de l'éclairage de la zone 2B en fonction de la présence et de l'éclairage naturel

- Sur la figure précédente, on voit que les deux premières rangées de luminaires sont allumées du fait que les capteurs de mouvement correspondant sont actifs.
- Bien que le capteur de mouvement relatif à la troisième rangée indique qu'il n'y a aucune présence, on voit que le troisième groupe de luminaires est allumé et ceci suite à une action sur le bouton poussoir.
- Les luminaires de la première rangée sont dimmables ; en effet, on utilise un capteur de lumière pour mesurer le niveau d'éclairement et adapter l'intensité d'éclairage des luminaires en fonction de l'apport de la lumière du jour pour avoir finalement le niveau d'éclairement requis. De ce fait, il est normal de voir que les luminaires de la première rangée sont moins éclairées que celles des deux autres rangées.

Le programme de gestion et régulation de l'éclairage du bureau est donné en **Annexe E**

3.2. Descriptif du système d'éclairage extérieur

Le système d'éclairage extérieur est commandé avec les même modes de gestion que celui de l'éclairage intérieur à une différence près, il comporte un nouveau mode de gestion, le mode *crépusculaire*, ce mode permet l'allumage de la lumière extérieur le matin dès que le niveau d'éclairement dépasse un certain seuil et l'extinction le soir dès que le niveau d'éclairement descend en dessous d'un certain seuil, pour cela on utilise un capteur crépusculaire dont le seuil est réglable.

La **figure IV.19** illustre le schéma électrique du tableau divisionnaire de l'éclairage extérieur, il comporte un disjoncteur de tête commandable ainsi que des disjoncteurs de départ pour la protection (ces disjoncteurs ne sont pas commandables, la commande se fait par action sur des télérupteurs ; non schématisés sur la figure)

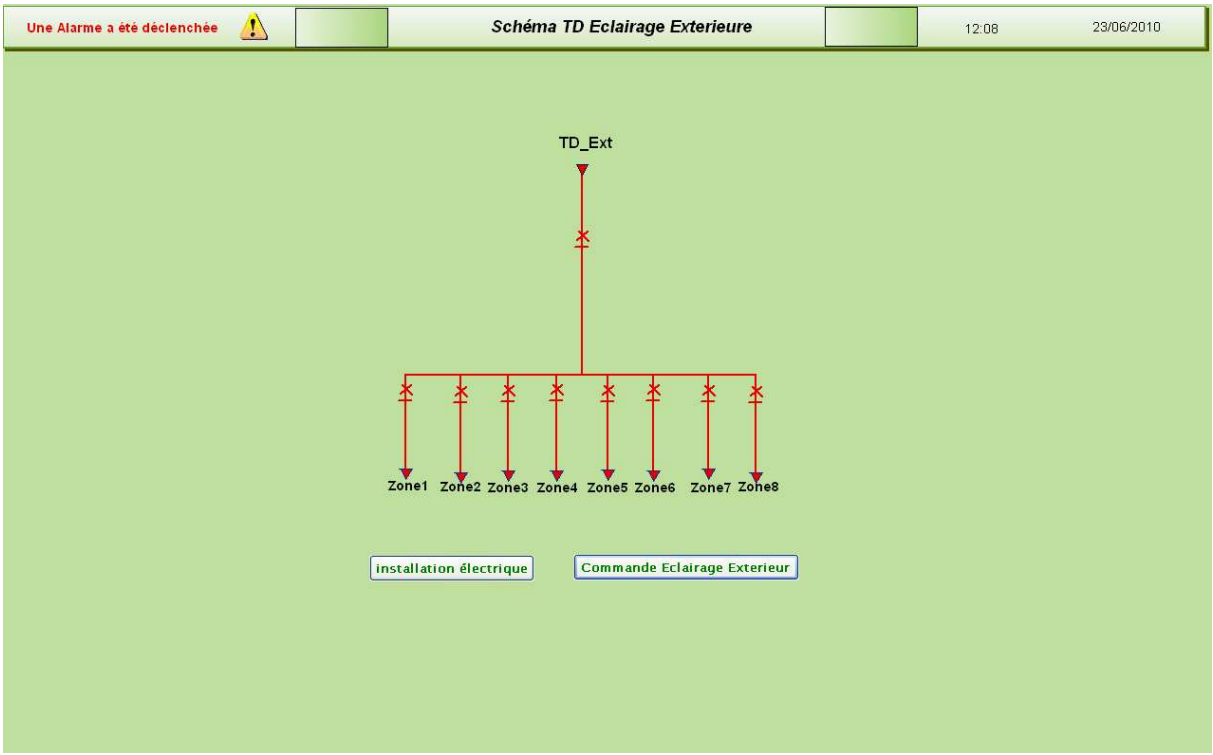


Fig. IV.19. Tableau divisionnaire de l'éclairage extérieur

La **figure IV.20** illustre la page de gestion de l'éclairage extérieur, on trouve les cases permettant de choisir l'un des trois modes de gestion, des boutons pour l'allumage et l'extinction des zones qui ne sont actives que dans le mode manuel, des voyant donnant l'état de l'éclairage des zones (allumées ou éteintes)

La page dispose aussi d'un espace permettant à l'opérateur de saisir l'heure d'allumage et d'extinction des zones configurées en mode horaire.

Par ailleurs, on a prévu des voyants pour indiquer l'état du disjoncteur de tête du tableau divisionnaire (alarmes arrivée), un autre voyant pour indiqué que l'un des disjoncteurs de protection a été déclenché (l'opérateur pourra déterminer le disjoncteur ouvert à partir des voyants indiquant l'état d'éclairage dans la page de gestion), finalement, on dispose aussi d'un voyant pour indiquer l'état du capteur crépusculaire, ce capteur (ayant une sortie logique) se met à 1 lorsque le niveau d'éclairement descend au dessous d'un seuil donné.

Les boutons de navigation permettent de se rendre soit vers la page d'accueil, soit vers le schéma d'installation électrique, ou encore vers le tableau divisionnaire de l'éclairage extérieur.

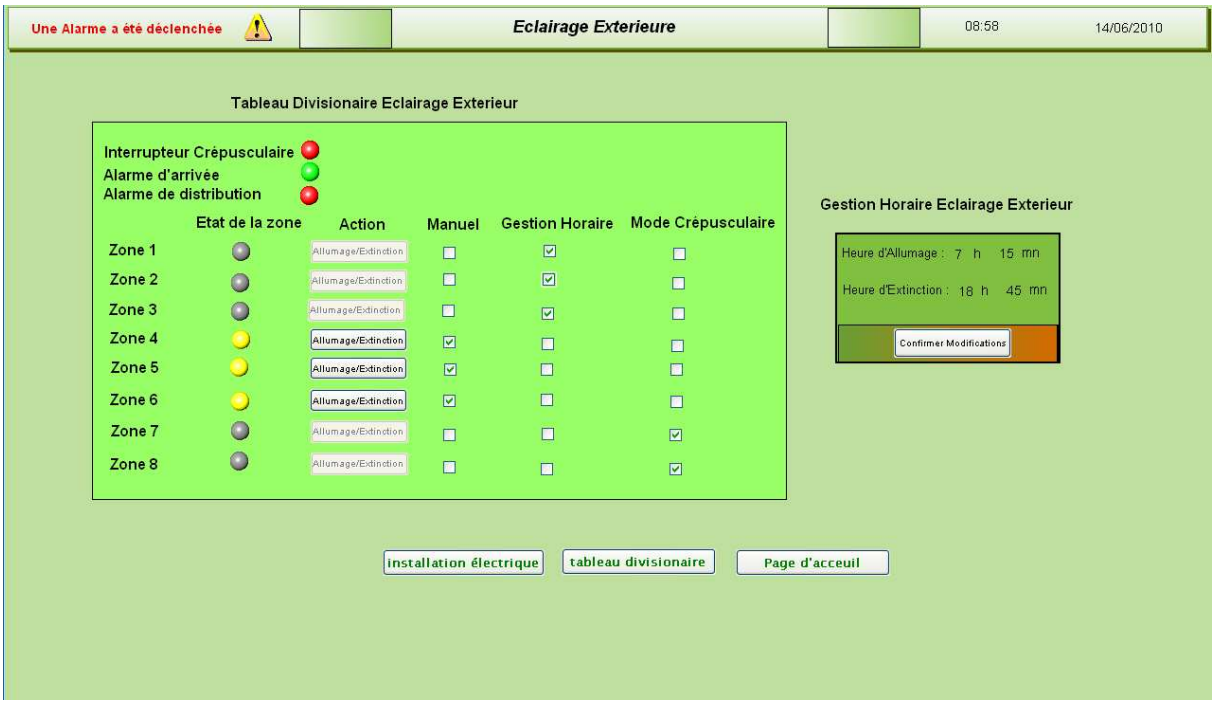


Fig. IV.20. Page de gestion de l'éclairage extérieur

4. Centrales de Traitement d'Air (CTA)

4.1 Présentation

Pour notre application, nous allons utiliser une CTA pour chaque deux niveaux, nous auront donc besoin de quatre CTA puisqu'il y a huit niveaux (Sous Sol, Rez de chaussée et les six étages).

Pour accéder à la page des CTA, il faut cliquer sur le bouton « Centrales de Traitement d'Air comme le montre la figure suivante :



Fig. IV.21. Accès à la page des CTA

La page des CTA nous permet d'accéder à l'une des quatre CTA comme le montre la **figure IV.22.**

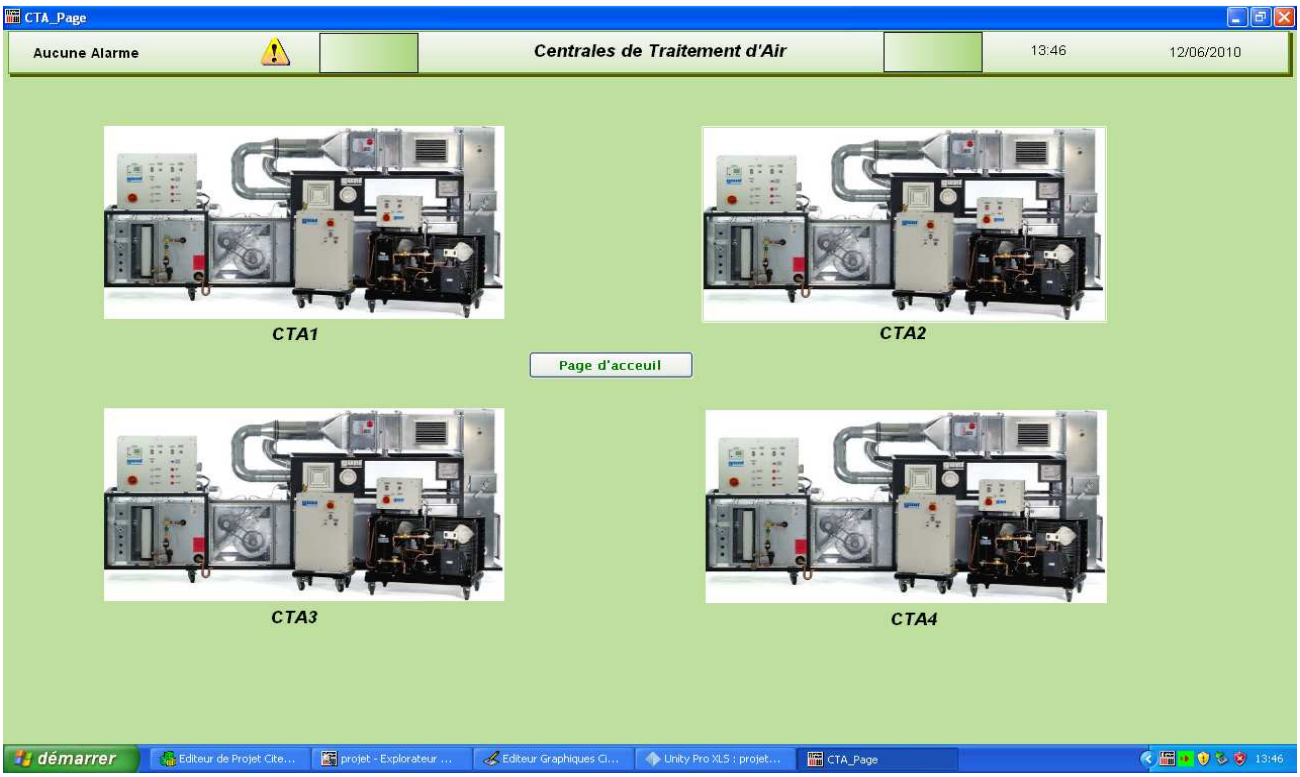


Fig. IV.22 Choix d'une CTA

En cliquant sur l'une des quatre CTA, on accède à une page représentant le synoptique de la CTA choisie afin de la configurer, d'entrer des consignes (débit du ventilateur, température d'air repris) et de visualiser certaines grandeurs.

En cliquant par exemple sur la figure « CTA1 », la page de la **figure IV.23** apparait

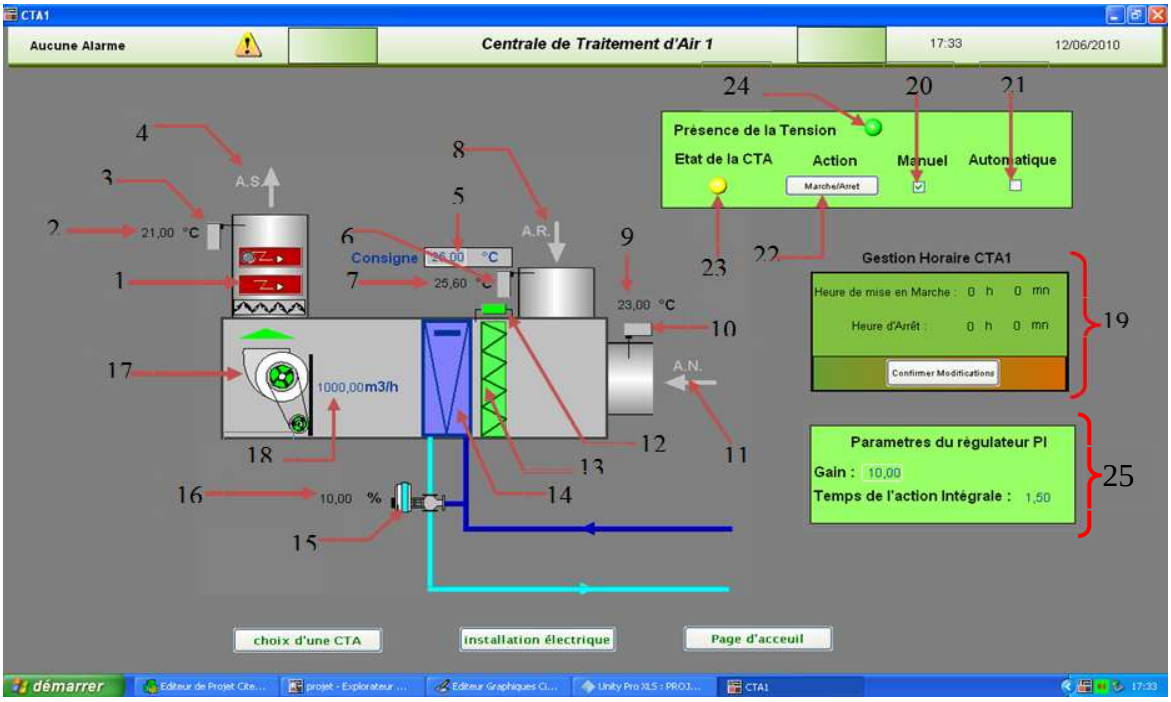


Fig. IV.23 Synoptique d'une CTA

1) Batterie chaude (électrique).
2) Affichage de la température d'air soufflé.
3) Capteur de température d'air soufflé.
4) Sortie d'air soufflé.
5) Entrée de la consigne « température d'air repris ».
6) Capteur de la température d'air repris.
7) Affichage de la température d'air repris.
8) Entrée de l'air repris.
9) Affichage de la température d'air neuf (extérieur).
10) Capteur de la température d'air neuf (extérieur).
11) Entrée de l'air neuf.
12) Capteur de pression différentiel servant à détecter l'encrassement du filtre.
13) Filtre à air.
14) Batterie d'eau froide.
15) Vanne 3 voies.
16) Taux d'ouverture de la vanne 3 voies (en %)
17) Ventilateur centrifuge.
18) Consigne de débit d'air soufflé par le ventilateur.

19) Entrée de l'heure de mise en marche et de l'arrêt de la CTA lorsqu'on est en mode « automatique » c.à.d. en mode gestion horaire de marche et d'extinction.
20) Mode manuel : mise en marche et extinction de la CTA grâce au bouton poussoir « Marche/Arrêt ».
21) Mode Automatique : mode de la gestion horaire.
22) Bouton « Marche/Arrêt » de la CTA
23) Etat de la CTA : en marche (couleur jaune) ou en arrêt (couleur grise).
24) Indique si la CTA est alimentée ou non.
25) Paramètres du régulateur PI.

Les différents capteurs et actionneurs utilisées dans les CTA sont détaillés en **Annexe D**.

4.2 Régulation de la température d'air repris

La régulation de la température d'air repris se fait en utilisant deux organes de réglages qui sont : la vanne 3 voies (pour modifier le débit d'eau froide arrivant à la batterie d'eau froide) et la batterie chaude électrique modulante à sortie Triac (de 0% à 100% de la puissance de chauffage fournie).

On voit bien sur le synoptique de la **figure IV.23** que c'est la température d'air soufflé qui sera influencée en premier par une variation sur l'une ou l'autre des commande. En effet la température d'air repris a une dynamique beaucoup plus lente que celle de la température d'air soufflé et ceci à cause des grands espaces que doit traverser l'air soufflé (gaines de soufflage) avant qu'il n'atteigne les différents locaux des deux étages (dont la CTA traite l'air).Il faut ajouter à cela la dynamique relativement lente des échanges thermiques dans les locaux.

Pour toutes ces raisons, il est préférable d'utiliser une régulation en cascade pour réguler la température d'air repris.

La boucle interne qui est la plus rapide est celle de la régulation de la température d'air soufflé en utilisant la vanne 3 voies et la batterie chaude électrique. On utilisera pour cela un régulateur **Split-Range.**

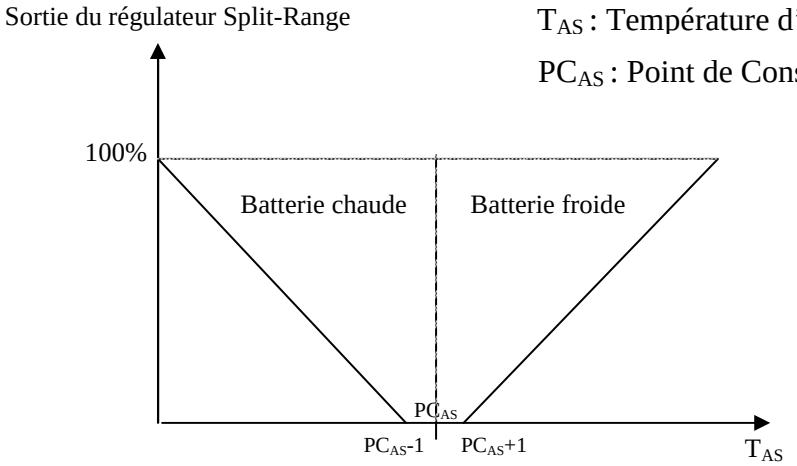


Fig. IV.24 Régulateur Split Range

La boucle externe (boucle maitre) quand à elle, est celle de la régulation de la température d'air repris. Un régulateur PI est prévu à cet effet dont la sortie est le point de consigne de la température d'air soufflé (PC_{AS}) qui sera limité en valeur inférieure (10°) et en valeur supérieur (40°).

Le schéma de régulation est illustré à la **figure IV.25**

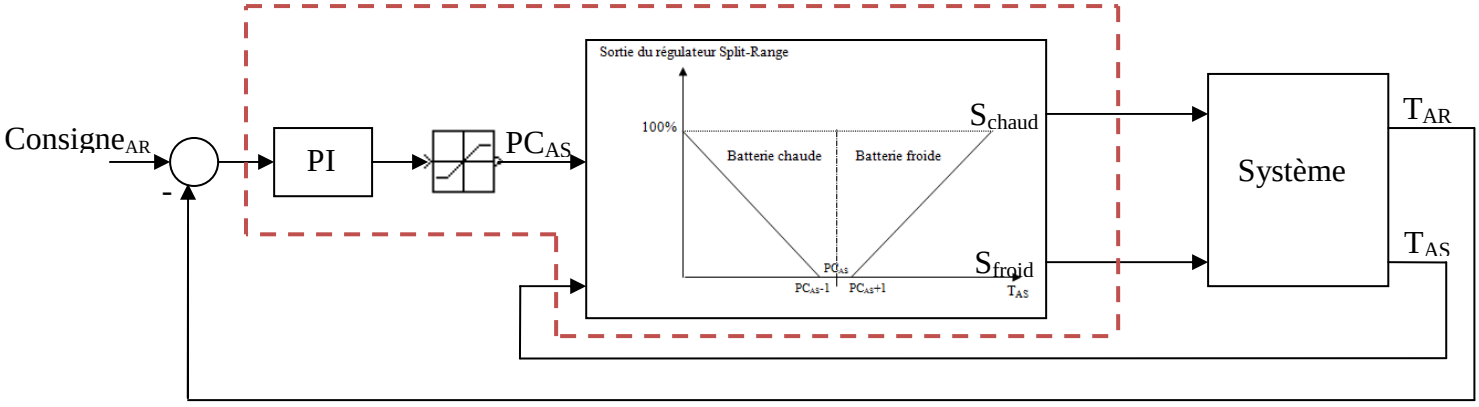


Fig. IV.25 Schéma de régulation de la CTA

- Consigne$_{AR}$ désigne ici la consigne de la température d'air repris.
- PC_{AS} désigne le point de consigne de la température d'air soufflé.
- T_{AR} est la température d'air repris mesurée.
- T_{AS} est la température d'air soufflé mesurée.
- S_{chaud} : c'est la sortie du Triac de la batterie chaude électrique en pourcentage de la puissance maximale.
- S_{froid} : c'est le taux d'ouverture de la vanne trois voies en %.
- Système : comprend les gaines de soufflage et d'extraction d'air ainsi que les différents locaux des deux étages.

Etant donné que la modélisation du comportement thermique du système est très complexe, les paramètres du régulateur PI devront être déterminés par des moyens expérimentaux.

4.3 Création d'une DFB pour la régulation de la température d'air repris de la CTA

Sous Unity Pro, nous avons crée une DFB représentant le régulateur de la **figure IV.25** (entre pointillé rouge). Cette DFB est représenté à la **figure IV.26**.

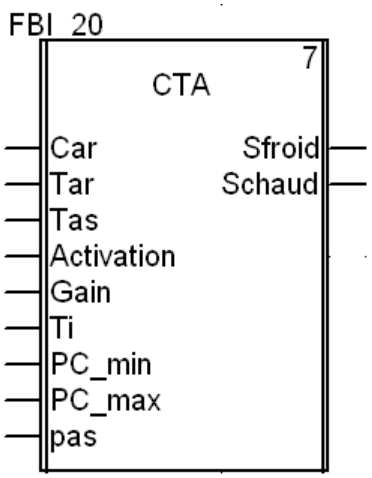


Fig. IV.26 DFB pour la régulation de la température d'air repris de la CTA

- **Activation** : Si ce bit est mis à 1 alors la régulation est activée.
- **Car** : Consigne de la température d'air repris (en °C)
- **Tar** : Température d'air repris mesurée (en °C)
- **Tas** : Température d'air soufflé mesurée (en °C)
- **pas** : représente le pas d'intégration qu'on prendra égal au temps de cycle.
- **PC_min** : Point de consigne de la température d'air soufflé (sortie du régulateur PI) minimale.
- **PC_max** : Point de consigne de la température d'air soufflé (sortie du régulateur PI) maximale.
- **Gain** : gain du régulateur PI.
- **Ti** : temps de l'action Intégrale (en s).
- **Schaud** : c'est la sortie du Triac de la batterie chaude électrique en pourcentage de la puissance maximale.
- **Sfroid :** c'est le taux d'ouverture de la vanne trois voies en %.

Pour programmer le régulateur PI en langage ST (Structured Text), nous avons approximé la fonction « intégrale » par la formule Riemann :

$$\int_a^b f(t)\,dt = \lim_{n\to\infty} \frac{b-a}{n} \sum_{k=0}^{n} f(a + k.\frac{b-a}{n})$$

En appliquant cette formule à l'erreur de mesure ε(t) avec a=0, b=t et n=$\frac{t}{\Delta t}$ on obtient :

$$\int_0^t \varepsilon(\tau)\,d\tau = \lim_{\Delta t\to 0} \Delta t \sum_{k=0}^{t/\Delta t} \varepsilon(k.\Delta t) \simeq \Delta t \sum_{k=0}^{t/\Delta t} \varepsilon(k.\Delta t)$$

On se contente donc de sommer l'ensemble des erreurs mesurées et en multipliant par le termeΔt.

Le programme pour la création de cette DFB est donné en **Annexe E.**

4.4 Gestion des alarmes de la CTA

Pour assurer le bon fonctionnement de la CTA, trois alarmes ont été prévu pour avertir l'opérateur de l'apparition d'un défaut CTA. Ces alarmes sont :

- **Surchauffe de la batterie chaude électrique** : Un thermostat de sécurité indique que la batterie chaude électrique a atteint une température critique (configurable).
- **Défaut ventilateur** : indique une défaillance du ventilateur de soufflage.
- **Filtre à air encrassé** : un capteur de pression différentielle indique dès dépassement d'un certain seuil l'encrassement du filtre à air.

Dès apparition de l'une des deux premières alarmes précédentes, le fonctionnement de la CTA est automatiquement interrompu jusqu'à disparition du défaut, cependant l'alarme persiste toujours jusqu'à ce qu'elle soit acquittée par l'opérateur.

L'alarme « Filtre à air encrassé » quand à elle n'interrompu pas le fonctionnement de la CTA car elle n'est pas critique.

4.5 Séquence de Marche/Arrêt de la CTA

La mise en **marche** de la CTA suit la séquence suivante :

1. Mise en service du ventilateur de soufflage.
2. Sur absence de défauts de la CTA, mise en service de la régulation.

La mise en **arrêt** de la CTA suit la séquence suivante :

1. Arrêt de la régulation.
2. Après une temporisation réglable arrêt du ventilateur de soufflage.

5. Inventaire des capteurs et actionneurs utilisés pour chaque niveau

Dans ce qui suit, nous allons donner l'inventaire des capteurs et actionneurs utilisés dans chacun des niveaux, et reliés par câblage au Momentum où à l'automate, afin de déterminer le nombre et le type de Momentum à utiliser.

Toutes les informations relatives aux équipements présentés dans cette section seront détaillées en **Annexe D**.

5.1 Sous Sol

5.1.1 Installation Electrique

Nom de l'équipement	Capteurs et Actionneurs associés	Total des Capteurs et Actionneurs associés
1 Masterpact NW6300 (Disjoncteur de tête du TGGE)	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	3
	Actionneurs :moteur réducteur 230V CA	1
3 Masterpact NW2500(tête des TGBT, couplage)	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	9
	Actionneurs :moteur réducteur 230V CA	3
1 Inverseur de source Masterpact NW 2500	**Capteurs** : sortie 230V AC de l'IVE	1
	Actionneur : boitier IVE commandable (230V CA)	1
2 Inverseur de source Masterpact NW 1000	**Capteurs** : sortie 230V AC de l'IVE	2
	Actionneur : boitier IVE commandable (230V CA)	2
5 Compact NS 1000	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	15
	Actionneurs :moteur réducteur 230V CA	5
5 Compact NSX 400	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	15
	Actionneurs : bloc de télécommande adaptable LV432641 230V CA	5
4 Compact NSX 250	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	12

	Actionneurs : bloc de télécommande adaptable LV432641 230V CA	4
30 Compact NSX 160	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	90
	Actionneurs : bloc de télécommande adaptable LV432641 230V CA	30
10 Disjoncteur C60N (départ vers chaque zone d'éclairage)	**Capteurs** : 1 OF 230V CA pour chaque disjoncteur.	10
10 télérupteur	Commandable par une tension 230V CA	10

Tableau. IV.1 inventaire de l'installation électrique située au sous sol

5.1.2 Centrale de Traitement d'Air

Cette centrale est commune au sous sol et au rez-de-chaussée.

Capteur	**Type**	**Total**
Sonde de température d'air neuf « TT 531 »	Analogique 0-10V	1
Sonde de température d'air soufflé et repris « KTM1-U »	Analogique 0-10V	2
Pressostat Différentiel « DPS »	TOR 230V AC	1
Thermostat pour la batterie chaude électrique.	TOR 230V AC	1
Actionneur		
Vanne mélangeuse 3 voies	Analogique 0-10V	1
Batterie chaude électrique	Analogique 0-10V	1

Tableau. IV.2 Capteur et actionneurs de la CTA 1 pour le Sous Sol et le rez-de-chaussée

5.1.3 Nombre de points d'Entrées/Sorties et choix des Momentum pour le Sous Sol

Les différents points d'E/S pour le sous sol sont résumés dans le tableau suivant :

Type de point	**Nombre**
Entrées TOR 230V CA	159
Entrées Analogiques 0-10V	2
Sorties TOR 230V CA	61
Sorties Analogiques 0-10V	2

Tableau. IV.3 Nombre et type de points d'E/S pour le Sous Sol

Nous auront ainsi besoin de Momentum ayant comme Embase les références et caractéristiques résumés dans le tableau suivant :

Référence de l'Embase du Momentum	Caractéristiques	Nombre
170 ADI 740 50	Embase d'entrée TOR 230V CA (**16 voies**)	10
170 AAI 140 00	Embase d'entrée analogique 0-10V CC (**16 voies**)	1
170 ADO 740 50	Embase de sortie TOR 230V CA (**16 voies**)	4
170 AAO 120 00	Embase de sortie analogique 0-10V CC (**4 voies**)	1

Tableau. IV.4 Choix des Embases de Momentum pour le Sous Sol

5.2 Rez-de-chaussée

5.2.1 Tableau divisionnaire d'éclairage

Nom de l'équipement	Capteurs et Actionneurs associés	Total des Capteurs et Actionneurs associés
1 Compact NSX 160	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	3
	Actionneurs : bloc de télécommande adaptable LV432641 230V CA	1
14 Disjoncteur C60N (départ vers chaque zone d'éclairage)	**Capteurs** : 1 OF 230V CA pour chaque disjoncteur.	14
14 télérupteur	Commandable par une tension 230V CA	14

Tableau. IV.5 inventaire des équipements du Tableau Divisionnaire d'Eclairage du Rez-de-chaussée

5.2.2 Nombre de points d'Entrées/Sorties et choix des Momentum pour le rez-de-chaussée

Les différents points d'E/S pour le rez-de-chaussée sont résumés dans le tableau suivant :

Type de point	Nombre
Entrées TOR 230V CA	17
Entrées Analogiques 0-10V	0
Sorties TOR 230V CA	15
Sorties Analogiques 0-10V	0

Tableau. IV.6 Nombre et type de points d'E/S pour le rez-de-chaussée

Nous auront ainsi besoin de Momentum ayant comme Embase les références et caractéristiques résumés dans le tableau suivant :

Référence de l'Embase du Momentum	Caractéristiques	Nombre
170 ADI 740 50	Embase d'entrée TOR 230V CA (**16 voies**)	2
170 ADO 740 50	Embase de sortie TOR 230V CA (**16 voies**)	1

Tableau. IV.7 Choix des Embases de Momentum pour le rez-de-chaussée

5.3 Premier Niveau

5.3.1 Tableau divisionnaire d'éclairage

Nom de l'équipement	Capteurs et Actionneurs associés	Total des Capteurs et Actionneurs associés
2 Compact NSX 160	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	6
	Actionneurs : bloc de télécommande adaptable LV432641 230V CA	2
17 Disjoncteur C60N (départ vers chaque zone d'éclairage)	**Capteurs** : 1 OF 230V CA pour chaque disjoncteur.	17
17 télérupteurs	Commandables par une tension 230V CA	17

Tableau. IV.8 inventaire des équipements du Tableau Divisionnaire d'Eclairage du premier niveau

5.3.2 Centrale de Traitement d'Air

Cette centrale est commune au premier et au deuxième étage

Capteur	Type	Total
Sonde de température d'air neuf « TT 531 »	Analogique 0-10V	1
Sonde de température d'air soufflé et repris « KTM1-U »	Analogique 0-10V	2
Pressostat Différentiel « DPS »	TOR 230V AC	1
Thermostat pour la batterie chaude électrique.	TOR 230V AC	1
Actionneur		
Vanne mélangeuse 3 voies	Analogique 0-10V	1
Batterie chaude électrique	Analogique 0-10V	1

Tableau. IV.9 Capteur et actionneurs de la CTA 2 pour le premier et le second niveau

5.3.3 Dispositifs d'éclairage de la Zone 2B du premier niveau

Capteur	**Type**	**Total**
Bouton poussoir	TOR 230V CA	3
Capteur de lumière SLR310	Analogique 0-10V	1
Capteur de mouvement PIR-TF-25	TOR 230V CA	3
Actionneur		
Ballast dimmable	Analogique 1-10V	3

Tableau. IV.10 Capteur et actionneurs pour l'éclairage de la Zone 2B

5.3.4 Nombre de points d'Entrées/Sorties et choix des Momentum pour le premier niveau

Les différents points d'E/S pour le premier niveau sont résumés dans le tableau suivant :

Type de point	**Nombre**
Entrées TOR 230V CA	31
Entrées Analogiques 0-10V	4
Sorties TOR 230V CA	19
Sorties Analogiques 0-10V	5

Tableau. IV.11 Nombre et type de points d'E/S pour le Premier niveau

Nous auront ainsi besoin de Momentum ayant comme Embase les références et caractéristiques résumés dans le tableau suivant :

Référence de l'Embase du Momentum	**Caractéristiques**	**Nombre**
170 ADI 740 50	Embase d'entrée TOR 230V CA (**16 voies**)	2
170 AAI 140 00	Embase d'entrée analogique 0-10V CC (**16 voies**)	1
170 ADO 740 50	Embase de sortie TOR 230V CA (**16 voies**)	2
170 AAO 120 00	Embase de sortie analogique 0-10V CC (**4 voies**)	2

Tableau. IV.12 Choix des Embases de Momentum pour le Premier niveau

5.4 Deuxième niveau

Contient les mêmes équipements que ceux présentés dans le **tableau IV.8**

Nous auront ainsi besoin de Momentum ayant comme Embase les références et caractéristiques résumés dans le tableau suivant :

Référence de l'Embase du Momentum	Caractéristiques	Nombre
170 ADI 740 50	Embase d'entrée TOR 230V CA (**16 voies**)	2
170 AAI 140 00	Embase d'entrée analogique 0-10V CC (**16 voies**)	0
170 ADO 740 50	Embase de sortie TOR 230V CA (**16 voies**)	2
170 AAO 120 00	Embase de sortie analogique 0-10V CC (**4 voies**)	0

Tableau. IV.13 Choix des Embases de Momentum pour le Second niveau

5.5 Troisième niveau

Contient les mêmes équipements que ceux présentés dans le **tableau IV.8** (pour l'éclairage) et **tableau IV.9** (pour la CTA commune aux niveaux 3 et 4)

Nous auront ainsi besoin de Momentum ayant comme Embase les références et caractéristiques résumés dans le tableau suivant :

Référence de l'Embase du Momentum	Caractéristiques	Nombre
170 ADI 740 50	Embase d'entrée TOR 230V CA (**16 voies**)	2
170 AAI 140 00	Embase d'entrée analogique 0-10V CC (**16 voies**)	1
170 ADO 740 50	Embase de sortie TOR 230V CA (**16 voies**)	2
170 AAO 120 00	Embase de sortie analogique 0-10V CC (**4 voies**)	1

Tableau. IV.14 Choix des Embases de Momentum pour le Troisième niveau

5.6 Quatrième niveau

Contient les mêmes équipements que ceux présentés dans le **tableau IV.8**

Nous auront ainsi besoin des mêmes équipements Momentum que ceux présentés dans le **Tableau IV.13**

5.7 Cinquième niveau

Contient les mêmes équipements que ceux présentés dans le **tableau IV.8** (pour l'éclairage) et **tableau IV.9** (pour la CTA commune aux niveaux 5 et 6).

Nous auront ainsi besoin des mêmes équipements Momentum que ceux présentés dans le **Tableau IV.14**

5.8 Sixième niveau

Contient les mêmes équipements que ceux présentés dans le **tableau IV.8**

Nous auront ainsi besoin des mêmes équipements Momentum que ceux présentés dans le **Tableau IV.13**

5.9 Zones extérieures du bâtiment

5.9.1 Tableau divisionnaire d'éclairage

Nom de l'équipement	Capteurs et Actionneurs associés	Total des Capteurs et Actionneurs associés
1 Compact NSX 160	**Capteurs** : 3 OF 230V CA pour chaque disjoncteur(1 par phase).	3
	Actionneurs : bloc de télécommande adaptable LV432641 230V CA	1
8 Disjoncteur C60N (départ vers chaque zone d'éclairage)	**Capteurs** : 1 OF 230V CA pour chaque disjoncteur.	8
8 télérupteurs	Commandables par une tension 230V CA	8

Capteur	Type	Total
Interrupteur crépusculaire	TOR 230V CA	1

Tableau. IV.15 Equipements pour les zones extérieures

5.9.2 Nombre de points d'Entrées/Sorties et choix des Momentum pour les zones extérieures

Les différents points d'E/S pour le premier les zones extérieures sont résumés dans le tableau suivant :

Type de point	Nombre
Entrées TOR 230V CA	12
Entrées Analogiques 0-10V	0
Sorties TOR 230V CA	9
Sorties Analogiques 0-10V	0

Tableau. IV.16 Nombre et type de points d'E/S pour les zones extérieures

Nous auront ainsi besoin de Momentum ayant comme Embase les références et caractéristiques résumés dans le tableau suivant :

Référence de l'Embase du Momentum	Caractéristiques	Nombre
170 ADI 740 50	Embase d'entrée TOR 230V CA (**16 voies**)	1
170 AAI 140 00	Embase d'entrée analogique 0-10V CC (**16 voies**)	0
170 ADO 740 50	Embase de sortie TOR 230V CA (**16 voies**)	1
170 AAO 120 00	Embase de sortie analogique 0-10V CC (**4 voies**)	0

Tableau. IV.17 Choix des Embases de Momentum pour les zones extérieures

6. Réseau de Communication du bâtiment

Les nombre total d'équipements distribués Momentum est résumé dans le tableau suivant :

Référence de l'Embase du Momentum	Caractéristiques	Nombre
170 ADI 740 50	Embase d'entrée TOR 230V CA (**16 voies**)	25
170 AAI 140 00	Embase d'entrée analogique 0-10V CC (**16 voies**)	4
170 ADO 740 50	Embase de sortie TOR 230V CA (**16 voies**)	18
170 AAO 120 00	Embase de sortie analogique 0-10V CC (**4 voies**)	5

Tableau. IV.17 Embases de Momentum pour tout le bâtiment

Tous ces Momentum communiquent via un réseau Ethernet Modbus TCP/IP à travers un Switch auquel sont reliés aussi l'automate programmable Quantum et la station de supervision.

Le synoptique de la partie communication est illustré par la **figure IV.27**

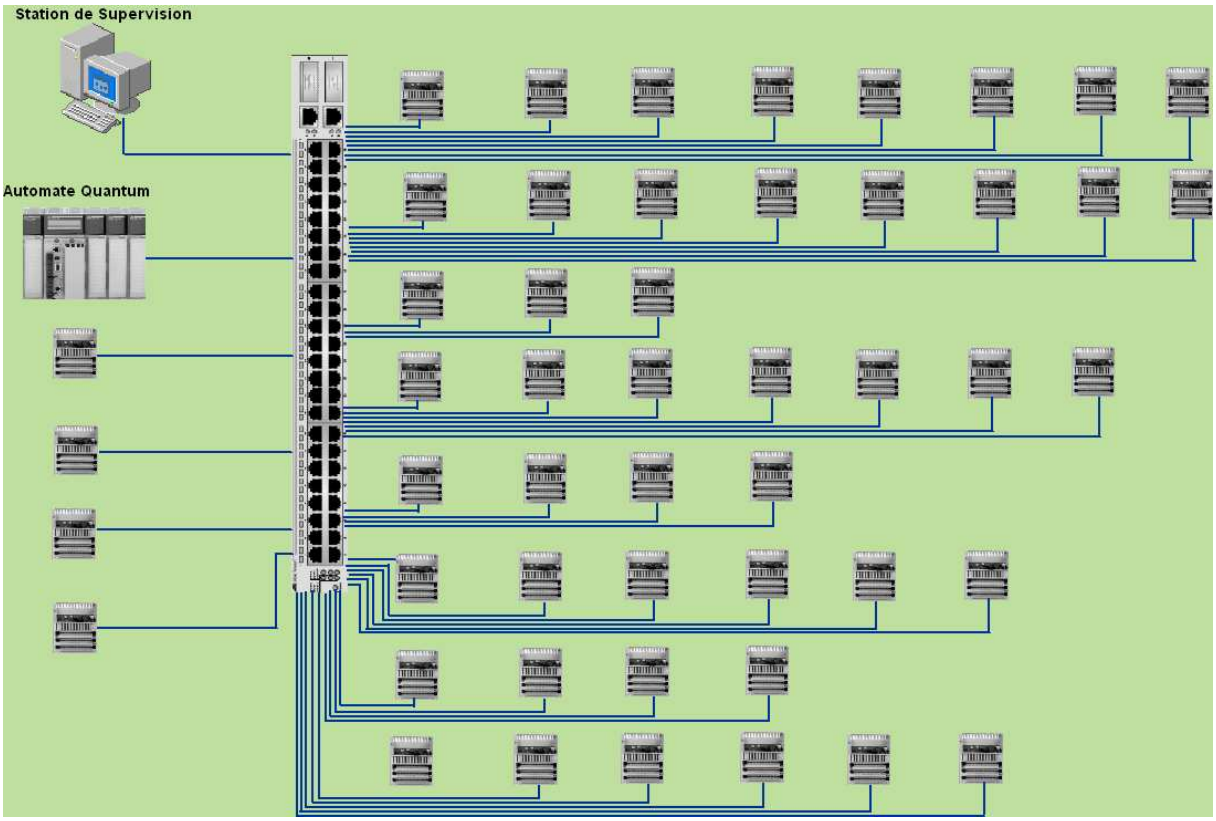


Fig. IV.27 Synoptique de la partie communication réseau

7. Conclusion

Durant ce chapitre, nous avons exposé l'application conçue sous Vijeo Citect pour la supervision de l'installation électrique, de l'éclairage et de climatisation pour un bâtiment à six niveaux avec rez-de-chaussée et sous sol. Pour se rapprocher au mieux de la réalité du terrain, nous avons prix soin de chercher dans diverses fiches techniques les équipements adéquats pour concevoir notre système de supervision.

Conclusion Générale

Conclusion Générale

Etant donné la place qu'occupe actuellement les bâtiments intelligents, leur consacrer un projet de fin d'étude nous a semblé une excellente manière d'appliquer nos connaissances acquises durant notre cycle d'ingénieur en automatique.

En effet, durant ce projet, nous avons conçu une application de supervision de l'installation électrique, d'éclairage et de climatisation d'un bâtiment et ceci en utilisant nos connaissances en automatique, électrotechnique et informatique industrielle. Nous avons de plus exploré divers domaines tels que le génie climatique qui traitent entre autre des théories du confort visuel et thermique. Ces domaine qui ne faisait pas partie de notre formation mais étaient indispensable à la conception de notre application.

Perspectives

Perspectives

Ce mémoire constitue l'aboutissement de notre cursus d'ingénieur mais également un point de départ pour d'autres projets de fin d'études traitant des bâtiments intelligents. En effet de nombreuses perspectives s'offrent à nous:

- Dimensionnement des équipements de l'installation électriques (disjoncteurs, fusibles…) et de la Centrale de Traitement d'Air (les différents capteurs et actionneurs, diamètres des conduits, type et puissance des ventilateurs…) en vue de l'implémentation de la station de supervision sur un bâtiment réel.
- Dimensionnement des paramètres des régulateurs utilisés et ceci soit par des méthodes expérimentales sur terrain ou par une modélisation analytique.
- Utilisation d'équipements intelligents et communiquant sur des réseaux orientés bâtiment.
- Intégration du GEG (Groupe d'Eau Glacée) dans le système de supervision.
- Donner des droits différents pour les opérateurs.

Bibliographie

Bibliographie

[B1] S. Wang, « Intelligent Buildings and Building Automation », Spon Press, 2010.

[B2] The Chartered Institution of Building Services Engineers, « Building Control Systems», Butterworth Heinemann, 2000.

[B3] **«** Document de formation pour une solution compléte d'automatisation »

[B4] W.C. Turner, S. Doty, « Energy Management Handbook », The Fairmont Press, 2006.

[B5] FEKNOUS Baya, HAMI Elias, « Development de blocs function sous UNITY PRO, Application aux Régulateurs P.I.D Auto-Ajustables », Projet de Fin d'Etude, Automatique, ENP 2010

[B6] FIHAKHIR Amine Mahdi, DALI Ali, « Gestion et Supervision d'une Station de Pompage à Base d'Automates Schneider à l'aide des DFB», Projet de Fin d'Etude, Automatique, ENP 2010

[B7] R.Mezari, M .Lahdir, « Mise en œuvre des réseaux locaux sous Linux et Windows », Pages Blues, 2007.

[B8] H. Merz, T. Hansemann, C. Hübner, « Building Automation : Communication Systems with EIB/KNX, LON and BACnet », Springer Series on Signal and Communication Technologies, 2009.

[B9] DEREUMAUX Pascal, « Réseaux Modbus », Novembre 2004

Sites web:

[W1] http://en.wikipedia.org

[W2] http://conseils.xpair.com/gestion_technique_batiment/conseil-consulter_savoir_faire-domotique_gtc_gtb-934.html

[W3] http://www.climatechange.org

[W4] http://www.schneider-electric.fr

[W5] Site Web CISCO

[W6] www.energieplus-lesite.be

Fiches Techniques :

[F1] Sauter components, « EGT 130 : Transmetteur de mesure de température ambiante », 2003

[F2] Sauter components, « SDU : Transmetteur faible pression différentielle», 2003

[F3] Sauter components, « EGH120 : Transmetteur d'humidité relative», 2003

[F4] Honeywell, « TT531 : Sonde de température extérieure», 2002

[F5] THERMASGARD, « TT531 : Sonde de température pour gaine d'air», 2002

[F6] Honeywell, « DPS : Pressostat Différentiel pour Ventilation et Climatisation», 1998

[F7] Karno, « Vannes de régulation trois voies», 2002

[F8] Schneider Electric, « Modicon Telemecanique Quantum PLC Platform», 2001

[F9] Hager, « Interrupteur Automatique 2/3 fils » 2009.

Manuels et Catalogues

[M1] Schneider Electric, « Catalogue résidentiel et petit tertiaire », 2010

[M2] Schneider Electric, « Efficacité Energétique des Bâtiments »,2009

[M3] Schneider Electric, « Guide de la distribution électrique basse tension et HTA »,2009

[M4] Schneider Electric, « Catalogue résidentiel et petit tertiaire »,2010

[M5] TAC Automation, « Automatisme du Bâtiment », 2007

الملخص

العمل المنجز في هذه المذكرة يتمحورحول تصميم نظام تحكم و مراقبة باستعمال البرنامج فيجبو سايتكة وهذا بهدف تسيير مختلف الأنظمة لبناية ذكية مثل النظام الكهربائي و نظام الاضاءة و نظام التكييف. البناية المعتبرة في هذا العمل تتكون من ستة طوابق بالاضافة الى طابق أرضي وطابق سفلي. المسير الصناعي المستعمل عبارة عن شنايدر كوانتوم حيث تم إعداده و برمجته باستعمال البرنامج يونيتي برو

الكلمات المفتاحية : فيجبو سايتكة, يونيتي برو، المسير الآلي الصناعي، التسيير التقني للبنايات، التسيير التقني للكهرباء التحكم بالإضاءة، الوحدة المركزية لتنقية الهواء

RESUME

Le travail présenté dans ce document consiste en la conception d'un système de supervision sous Vijeo Citect de l'installation électrique, de l'éclairage et de la climatisation pour un bâtiment à six étages avec sous sol et rez-de-chaussée. L'automate programmable utilisée est un Quantum qui fait partie de la gamme des automates Programmables Schneider Electric et que nous avons programmé et configuré par le logiciel Unity Pro.

Mot clés : Vijeo Citect, Unity Pro, Automate Programmable Industriel, Gestion Technique du Bâtiment, Gestion Technique d'Electricité, Gestion d'Eclairage, Centrale de Traitement d'Air.

Abstract

The work presented in this paper aims for the design of a supervisory application using the SCADA software Vijeo Citect, this application allows the control and supervision of various systems such as the electrical system, the lighting and the chilling system all belonging to a building which consists of six levels plus a basement and a ground floor, the plc used is a Quantum which belongs to Schnieder's plc range. The configuration and programming of this programmable logic controller is done using the software Unity Pro

Keywords : Vijeo Citect, Unity Pro, Programmable Logic Controller, Building Management Systems, Electrical Management Systems, Lighting Control, Air Handling Unit.

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



ECOLE NATIONALE POLYTECHNIQUE

Département d'Automatique
Spécialité Automatique

Projet de fin d'études

Pour l'obtention du diplôme
D'Ingénieur d'Etat en Automatique

THÈME

Conception d'un système de supervision pour un bâtiment intelligent :

Gestion de la distribution électrique, de l'éclairage et de la climatisation

ANNEXES

Etudié par :

DADDI HAMMOU Abdennour
RAHMANI Mustapha Amine

Proposé et dirigé par :

M^{r}. A. BERKOUK
M^{r}. M.A. MESSAIF

Juin 2010

Ecole Nationale Polytechnique, 10, AV. Hassen Badi, El-Harrach, Algérie

ANNEXE A

Présentation de la gamme d’automates Schneider

1. Introduction

Au cours de ce chapitre, nous allons exposer brièvement l'offre de Schneider Electric en automates programmables industriels (API) et en module d'E/S déportés Momentum, on s'étalera un peu plus avec l'automate que nous allons utiliser pour notre application, à savoir le Quantum.

2. La gamme Schneider Electric en automates programmables

Schneider Electric possède une large gamme d'automates qui diffèrent par leurs capacités de calcul et le nombre d'E/S offrant à l'utilisateur ainsi le choix de la gamme adéquate selon la complexité de l'application.

2.1. ZELIO LOGIC

C'est un module logique programmable destiné à la réalisation de petits automatismes de moins 40 E/S pour des applications telles que :

- comptage de pièces
- commande de pompes et de compresseurs
- convoyeurs
- panneaux d'affichage
- volets roulants.

Sa programmation, très intuitive, s'effectue directement sur la face ou à l'aide du logiciel Zelio Soft. Existe en version sans afficheur ni touches.

Les principales caractéristiques du ZELIO sont :

- gamme compacte : 3 modèles monoblocs de 10, 12, 20 E/S (avec ou sans afficheur et touches) ;
- gamme modulaire : 2 bases de 10 ou 26 E/S extensibles jusqu'à 40 E/S par modules d'extension de 6, 10 ou 14 E/S et un module de communication Modbus ;
- alimentation : 100/240 VCA ou 24 VCC ;
- fixation sans accessoire avec nez modulaire de 45 mm ;
- blocs fonctions prêts à l'emploi ;
- logiciel de programmation sur PC.



Fig. A.1. Automate ZELIO [B6]

2.2. TWIDO

Version compacte ou modulaire partage des options, des extensions d'E/S et un logiciel de programmation communs. Application : installations simples et unitaires, machines répétitive et compactes.
Les principales caractéristiques sont :

- Applications standards de 10 à 100 E/S (maximum 252 E/S), 2 types de bases : compactes de 6, 10 ou 24 E/S et modulaires de 20 ou 40 E/S ;
- Extension d'E/S : 14 modules TOR et 4 analogiques ;
- Raccordement : borniers à vis ou à ressort, précâblage : connecteur HE10, Twido Fast, déport des E/S, maître AS-interface ;
- Options : mémoire, horodateur, communication (liaison série), réglage (afficheur)
- Compacité : 40 E/S pour 95x90x70 mm
- Fonctionnalités : compteurs rapides, sorties impulsionnelles, 1 port RS 485 multiprotocoles : Modbus Maître/esclave, ASCII, traitement rapide sur événements, type de donnés (mots, double mots, arithmétique flottante), jusqu'à 14 boucles PID…



Fig. A.2. TWD RDA 10 RDF [B6]

2.3. TSX MICRO

C'est un Automate développé pour satisfaire au mieux les exigences d'adaptabilité et de maintenabilité des machines. Sa modularité et sa compacité répondent de manière économique à l'automatisation aussi bien de machines simples à quelques dizaines d'E/S que de machines plus complexes à 480 E/S. Pour simplifier le câblage des machines, le TSX MICRO supporte la connexion du bus AS-i. il se programme à l'aide du logiciel PL7 Micro sous Windows.

Les principales caractéristiques sont:

- Jusqu'à 248 E/S TOR « in rack »;
- 96 E/S par déport TSX Nano ;
- 248 E/S sur bs AS ;
- Fonctions de comptage rapide (50kHz), analogique et régulation ;
- Sécurité machine ;
- Communication Uni-Telway, ASCII, Modbus, Fipway, FIPIO agent, modem.



Fig. A.3 Automate TSX 35 05 [B6]

2.4. PREMIUM

Cette plate-forme d'automatismes, grâce à son architecture distribuée permet de répartir les cartes d'E/S et les fonctions métiers au plus près de l'application. Idéale pour les machines et installations modulaires et réparties, la gamme TSX Premium se décline :

- En format standard (TSX).
- EN format co-processeur pour PC.

L'intégration logicielle et matérielle des métiers permet de diminuer les temps de développement et de mise en route de l'installation. Enfin TSX Premium s'intègre à tous les niveaux de l'architecture de communication. Il se programme à l'aide des logiciels Unity Pro ou PL7.

Ses principales caractéristiques sont :

- Jusqu'à 2048 E/S TOR « in rack »;
- E/S à distance sur bus FIPIO et tiers ;
- 8 coupleurs AS-i, 248 E/S sur bus AS-I;
- Fonction de comptage rapide (1 MHz), analogique et régulation intégrée par configuration ;
- Sécurité machine ;
- Commande de mouvement multiaxe ;
- Communication Uni-Telway, ASCII, Modbus, Fipway,FIPIO, Ethernet;
- TCP-IP, Ethway;

2.5. Présentation de la Plate-forme Quantum

Le système Quantum est un calculateur spécifique offrant des fonctions de traitement numérique. Quantum est conçu pour fournir un contrôle en temps réel aux applications industrielles et de production dans une architecture modulaire et évolutive utilisant les modules suivants :

- Modules de l'automate (CPU)
- Modules d'alimentation (CPS)
- Modules d'E/S (Dxx, Axx)
- Modules d'interface réseau (y compris les modules bus de terrain)
- Modules intelligents/spécifiques



Fig. A.4. Automate Quantum [B6]

Il offre ainsi de nombreuses fonctionnalités dont on peut citer :

- Temps de cycle réduit avec acquisition rapide des entrées/sorties.
- Traitement d'interruption sur événement de temps ou en provenance d'entrées.
- Traitement possible en tâche rapide comme en tâche maître.
- Extension des capacités mémoire par cartes mémoire PCMCIA.
- Multiples ports de communication intégrés au processeur.
- Diagnostic et maintenance aisés grâce au bloc de visualisation LCD en face avant des processeurs haut de gamme.

2.5.1. L'Unité Centrale (CPU)

Le module est installé sur un rack d'E/S locales Quantum. Il s'agit d'un système électronique numérique qui utilise une mémoire programmable pour le stockage interne des instructions de l'utilisateur. Ces instructions servent à implémenter des fonctions spécifiques, telles que: l'arithmétique, la logique, le couplage…etc.

Quantum possède deux types d'UC : les UC de base et les UC avancées qui ont les caractéristiques suivantes :

	Processeur	SRAM (octets)	Programme IEC maximum	Programme IEC maximum (avec PCMCIA)	Interrupteur à clé
De base	140 CPU 311 10	2 Mo	400 Mo	Sans objet	Non
	140 CPU 434 12•	2 Mo	800 Mo	Sans objet	Oui
	140 CPU 534 14•	4 Mo	2,7 Mo	Sans objet	Oui
Avancé	140 CPU 651 50	2 Mo	512 Mo	7 168 Mo	Oui
	140 CPU 651 60	2 Mo	1 024 Mo	7 168 Mo	Oui
	140 CPU 651 60S (Sécurité)	2 Mo	1 024 Mo	7 168 Mo	Oui
	140 CPU 652 60	4 Mo	3 072 Mo	7 168 Mo	Oui
	140 CPU 671 60 (Redondance d'UC)	2 Mo	1 024 Mo	7 168 Mo	Oui
	140 CPU 671 60S (Redondance d'UC et sécurité)	2 Mo	1 024 Mo	7 168 Mo	Oui

Tableau A.1 Caractéristiques des UC Quantum

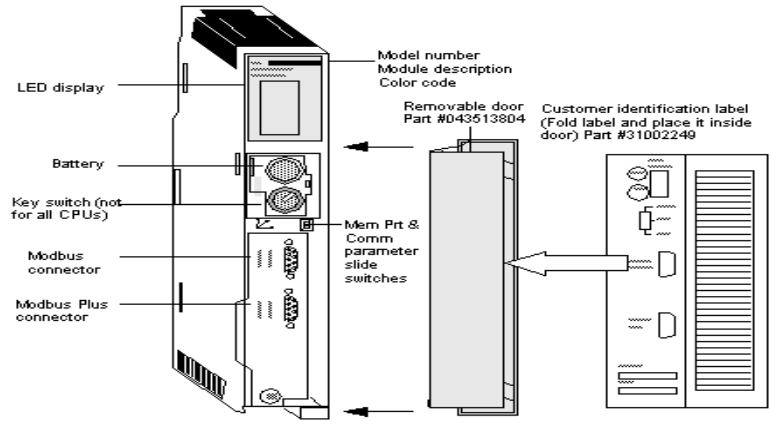


Fig. A.5. Processeur de base

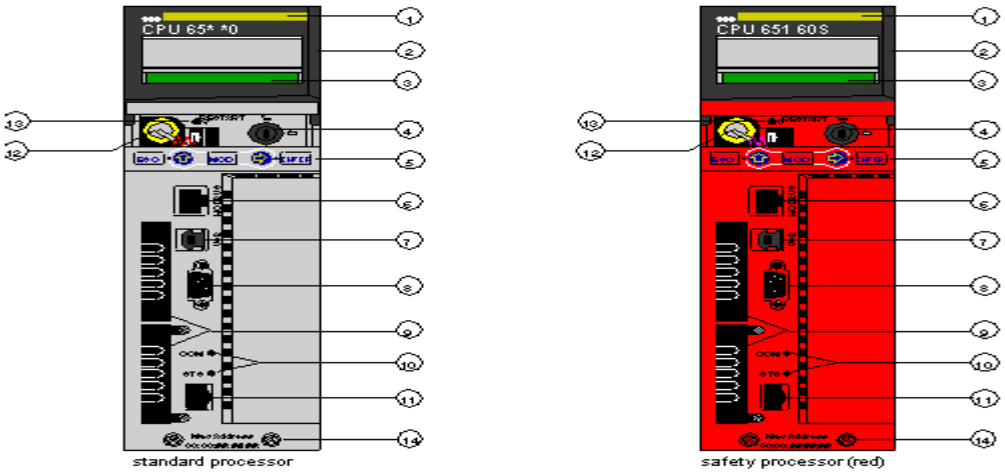


Fig. A.6 Processeurs avancés

1. Numéro du modèle, description du module, code couleur
2. Couvre-objectif (ouvert)
3. Ecran LCD (recouvert ici par le couvre-objectif)
4. Interrupteur à clé
5. Clavier (comportant 2 voyants rouges)
6. Port Modbus (RS-232) (RS-485)
7. Port USB
8. Port Modbus Plus
9. Emplacements PCMCIA (A et B)
10. Voyants (jaunes) pour la communication Ethernet
11. Port Ethernet
12. Pile (installée par l'utilisateur)
13. Bouton de réinitialisation
14. 2 vis

2.5.2. Modules d'alimentation (CPS)

Il existe d'importantes différences de conception entre les divers modèles d'alimentations Quantum et les concepteurs doivent en tenir compte afin d'optimiser les performances des systèmes.

Les alimentations Quantum comportent une logique intégrée de détection rapide des problèmes d'alimentation qui permet de signaler toute perte au niveau de la puissance d'entrée à l'ensemble des autres modules du rack. Ce signal, appelé POK (power OK), est actif à l'état haut. Le signal POK est représenté par le voyant Pwr ok (diode électroluminescente), situé sur le panneau avant de toutes les alimentations. Il est généré afin d'assurer un laps de temps suffisant entre le front négatif du POK système (problème d'alimentation) et la coupure effective de l'alimentation du rack. Cette première notification de coupure électrique est nécessaire pour que l'exécutif Quantum puisse procéder à un arrêt ordonné du système.

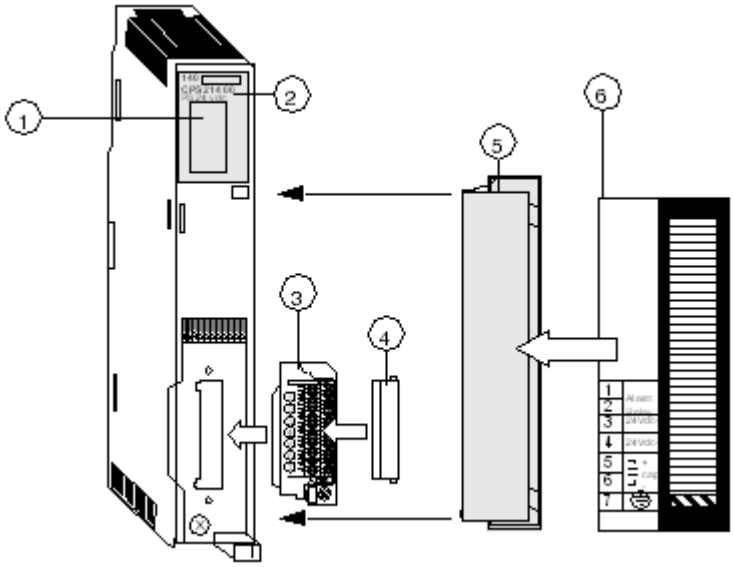


Fig. A.7. Vue d'un module d'alimentation

1. Voyants
2. Numéro du modèle, description du module, code couleur
3. Connecteur de câblage
4. Capot de protection du connecteur de câblage
5. Capot amovible
6. Étiquette d'identification.

Le tableau suivant donne les différentes alimentations existantes pour l'automate Quantum ainsi que leurs caractéristiques :

Type	Références	Tension source	Courant bus
Autonome	140 CPS 111 00	115 ... 230 V ca	3 A
	140 CPS 114 00	115/230 V ca	8 A
	140 CPS 211 00	24 V cc	3 A
	140 CPS 511 00	125 V cc	3 A
Autonome/Sommable	140 CPS 114 10	115/230 V ca	8 A
	140 CPS 114 20	115/230 V ca	11 A
	140 CPS 214 00	24 V cc	8 A
	140 CPS 414 00	48 ... 72 V cc	8 A
Autonome/Redondant	140 CPS 124 00	115/230 V ca	8 A
	140 CPS 124 20	115/230 V ca	11 A
	140 CPS 224 00	24 V cc	8 A
	140 CPS 424 00	48 ... 72 V cc	8 A
	140 CPS 524 00	125 V cc	8 A

Tableau. A.2. Alimentations pour automate Quantum

2.5.3 Sauvegarde du contenu de la mémoire

Dans les CPU Quantum, le contenu de la RAM est sauvegardé par l'intermédiaire d'une pile de sauvegarde logée en face avant du processeur.

Pour protéger le programme utilisateur, les processeurs sont aussi équipés d'un commutateur à clef (pour le verrouillage et le déverrouillage de l'accès à la mémoire programme).

3. EQUIPEMENT DISTANT MOMENTUM

3.1. Présentation du Momentum

La décentralisation des entrées/sorties permet, tout en ayant des performances comparables à celles obtenues dans une architecture centralisée, de répondre aux mieux aux besoins des utilisateurs et des constructeurs de machines :

- limitation du volume de câblage lié aux capteurs et pré-actionneurs.
- suppression des contraintes mécaniques liées aux chemins de câbles.
- réduction des temps d'étude et de test de la connectique.
- flexibilité et évolutivité des installations

Parmi les équipements distants les plus répandus, on trouve le Momentum. Ce dernier comprend quatre composants fondamentaux qui s'emboitent facilement :

- Adaptateurs de communication
- Embase d'E/S
- Adaptateurs pour processeur
- Adaptateurs optionnels

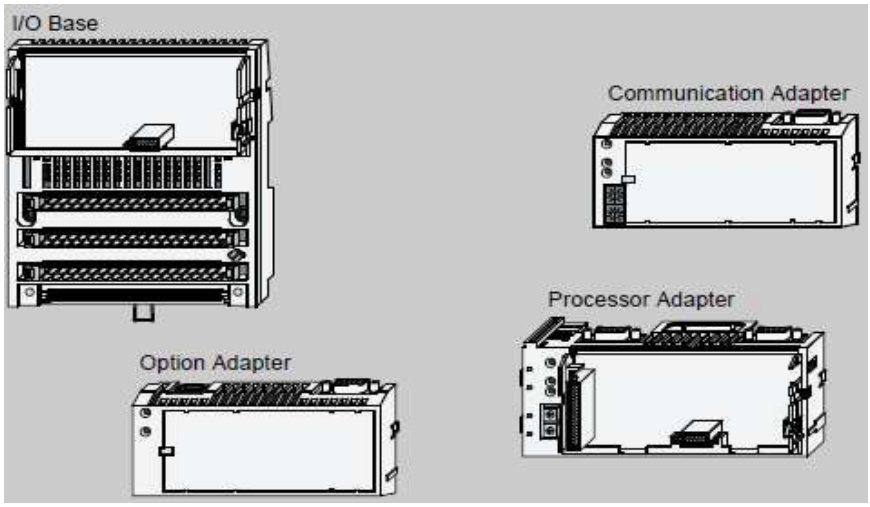


Fig. A.8. Les quatre composants fondamentaux du Momentum

3.2. Adaptateurs de communication

Dans le Momentum, l'embase et l'adaptateur de communication sont fournis séparément offrant ainsi une grande flexibilité et une adaptation à un grand nombre de réseaux par le choix de l'adaptateur de communication adéquat.

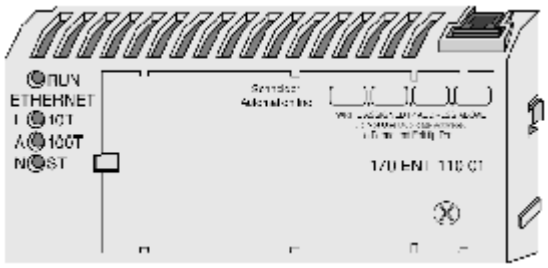


Fig. A.9.Adaptateur de communication Ethernet du Momentum

3.3. Embase d'E/S

C'est à cette embase que seront reliés, au travers de borniers, les différents capteurs et actionneurs distribués.

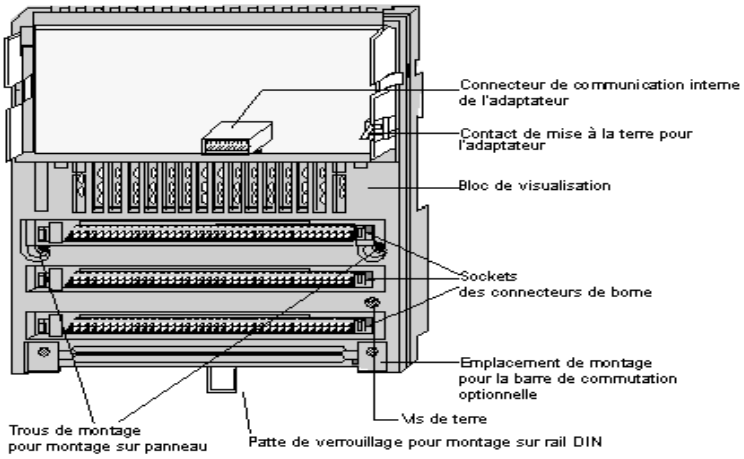


Fig. A.10. Embase d'E/S

- **Connecteur de communication interne :** Le connecteur de communication interne d'embase d'E/S fournit une communication automatique à tout adaptateur monté sur l'embase.
- **Bloc de visualisation :** Chaque embase possède un bloc de visualisation différent qui fournit des informations sur le statut des équipements d'entrée et de sortie.
- **Contact de mise à la terre :** Ce contact fournit une connexion à la terre à tout adaptateur monté sur l'embase.
- **Sockets des connecteurs de borne** : Chaque embase est équipée de sockets pour jusqu'à trois connecteurs de borne. Ceux-ci sont nécessaires à la connexion d'équipements d'E/S et doivent être commandés séparément.
- **Emplacement de la barre de commutation :** L'emplacement situé au bas de l'embase permet de brancher une barre de commutation pour prendre en charge les appareils à 3 et 4 fils. Les barres de commutation sont en option. Elles doivent être commandées séparément.
- **Montage :** Chaque embase possède des trous de montage pour un montage sur panneau et une patte de verrouillage pour le montage d'un rail DIN.

Les différents types d'embases sont résumés par les tableaux suivants :

Référence	Voies	Type	Détails
170 AAI 030 00	8	Entrée	Détection de rupture de ligne
170 AAI 140 00	16	Entrée	Asymétrique
170 AAI 520 40	4	Entrée	RTD/Thermocouple/mV
170 AAO 120 00	4	Sortie	0 à 20 mA
170 AAO 921 00	4	Sortie	4 à 20 mA

Tableau. A.3. Embases analogiques

Référence	Points	Type	Détails
170 ADI 340 00	16	Entrée	24 Vcc
170 ADI 350 00	32	Entrée	24 Vcc
170 ADI 540 50	16	Entrée	120 Vca
170 ADI 740 50	16	Entrée	230 Vca
170 ADM 350 10	16 entrées 16 sorties	Entrée Sortie	24 Vcc, True High
170 ADM 350 11	16 entrées 16 sorties	Entrée Sortie	24 Vcc, True High Entrées rapides
170 ADM 350 15	16 entrées 16 sorties	Entrée Sortie	24 Vcc, True Low
170 ADM 370 10	16 entrées 8 sorties	Entrée Sortie	24 Vcc à 2 A
170 ADM 390 10	16 entrées 12 sorties	Entrée Sortie	24 Vcc
170 ADM 390 30	10 entrées 8 sorties à relais	Entrée Sortie	24 Vcc
170 ADM 690 51	10 entrées 8 sorties	Entrée Sortie	120 Vca
170 ADO 340 00	16	Sortie	24 Vcc
170 ADO 350 00	32	Sortie	24 Vcc
170 ADO 530 50	8	Sortie	115 Vca à 2 A
170 ADO 540 50	16	Sortie	120 Vca
170 ADO 730 50	8	Sortie	230 Vca à 2 A
170 ADO 740 50	16	Sortie	230 Vca
170 ARM 370 30	10 entrées 8 sorties	Entrée Sortie	Alimentation 120 Vca Entrée 24 Vcc

Tableau. A.4. Embases TOR

Référence	Voies	Type	Détails
170 AMM 090 00	4 entrées analogiques 2 sorties analogiques 4 entrées TOR 2 sorties TOR	Entrée/Sortie	24 Vcc
170 ANR 120 90 unipolaire	6 entrées analogiques 4 sorties analogiques 8 entrées TOR 8 sorties TOR	Entrée/Sortie	24 Vcc
170 ANR 120 91 bipolaire	6 entrées analogiques 4 sorties analogiques 8 entrées TOR 8 sorties TOR	Entrée/Sortie	24 Vcc

Tableau. A.5. Embases mixtes

3.4. Processeurs Momentum

Quand une intelligence distribuée est requise à un point de control éloigné, des processeurs sont alors nécessaires pour un traitement local des informations. Les adaptateurs pour processeurs Momentum sont des PLC contenant une CPU, une RAM et une mémoire flash. Ces adaptateurs s'intègrent aux embases d'E/S de la même manière qu'un adaptateur de communication.

Les processeurs Momentum disponibles sont résumés dans le tableau suivant:

Modèle	Mémoire interne	RAM Flash	Vitesse d'horloge	Ports de communication
171 CCS 700 00	64 Ko	256 Ko	20 MHz	Un port Modbus RS-232
171 CCS 700 10	64 Ko	256 Ko	32 MHz	Un port Modbus RS-232
171 CCS 760 00	256 Ko	256 Ko	32 MHz	Un port Modbus RS-232
				Un port de bus d'E/S
171 CCC 760 10	512 Ko	512 Ko	32 MHz	Un port Modbus RS-232
				Un port de bus d'E/S
171 CCS 780 00	64 Ko	256 Ko	20 MHz	Un port Modbus RS-232
				Un port Modbus RS-485
171 CCC 780 10	512 Ko	512 Ko	32 MHz	Un port Modbus RS-232
				Un port Modbus RS-485
171 CCC 960 20	512 Ko	512 Ko	50 MHz	Un port Ethernet
				Un port de bus d'E/S
171 CCC 960 30	512 Ko	512 Ko	50 MHz	Un port Ethernet
				Un port de bus d'E/S
171 CCC 980 20	512 Ko	1 Mo	50 MHz	Un port Ethernet
				Un port Modbus RS-485
171 CCC 980 30	512 Ko	1 Mo	50 MHz	Un port Ethernet
				Un port Modbus RS-485
171 CCC 960 91	512 Ko	512 Ko	50 MHz	Un port Ethernet
				Un port de bus d'E/S
171 CCC 980 91	512 Ko	1 Mo	50 MHz	Un port Ethernet
				Un port Modbus RS-485
171 CBB 970 30	512 Ko	1 Mo	50 MHz	Quatre ports Ethernet
				Un port Modbus RS-232/485

Tableau. A.6. Processeurs Momentum

3.5 Configuration de l'adaptateur de communication Ethernet Momentum 170ENT11001

Etant donné que notre application s'articule autour du réseau Ethernet, nous avons choisi de prendre l'adaptateur de communication Ethernet Momentum 170ENT11001.

Le Momentum 170ENT11001 étant un adaptateur de communication Ethernet et sachant qu'à chaque équipement Ethernet correspond une adresse MAC unique. Par conséquent, une adresse MAC unique est affectée à chaque adaptateur Momentum 170ENT11001.

En règle générale, les équipements Ethernet sont désignés par une adresse IP. Par conséquent, l'équipement requiert une adresse IP pour pouvoir fonctionner sur un réseau local. Lorsque l'adaptateur Momentum 170ENT11001 est mis sous tension pour la première fois, l'équipement obtient une adresse IP à partir d'un des éléments suivants :

- le serveur BOOTP/DHCP possédant une adresse IP
- L'adresse IP sauvegardée dans la mémoire flash de l'adaptateur.

- l'adresse IP par défaut (dérivée de l'adresse MAC)

Lors de la première utilisation, après l'obtention d'une adresse IP du serveur BOOTP/DHCP, l'équipement dispose d'un ensemble de paramètres d'exécution par défaut et les E/S sont opérationnelles. Si les paramètres IP par défaut sont utilisés, l'équipement ne possède pas de paramètres d'exécution par défaut et les E/S ne sont pas opérationnelles.

Les paramètres d'exécution contiennent :

- les paramètres de l'adaptateur de communication.
- les paramètres de configuration SNMP.
- les paramètres de configuration de sécurité.

Il est recommandé de configurer les paramètres d'exécution ainsi que les paramètres IP.

Au démarage, l'équipement Momentum effectue les opérations suivantes pour determiner son adresse IP :

Si...	**Alors...**
l'équipement est sous tension,	l'équipement envoie un message au serveur BOOTP/DHCP pour demander une adresse IP. • L'équipement utilise l'adresse IP provenant d'un serveur BOOTP. • L'équipement vérifie les paramètres de configuration (exécution). ◦ Si les paramètres de configuration (exécution) **existent** dans la mémoire Flash, l'équipement utilise les paramètres de configuration (exécution). L'équipement est opérationnel. ◦ Si les paramètres de configuration (exécution) **n'existent pas** dans la mémoire Flash, l'équipement utilise les paramètres de configuration par défaut et est opérationnel. **Remarque** : Les paramètres par défaut ne sont pas forcément adaptés à votre système.
le serveur BOOTP/DHCP ne fournit pas d'adresse IP,	l'équipement recherche l'adresse IP configurée enregistrée dans la mémoire Flash. • L'équipement utilise l'adresse IP enregistrée dans la mémoire Flash. • L'équipement vérifie les paramètres de configuration (exécution). ◦ Si les paramètres de configuration (exécution) **existent** dans la mémoire Flash, l'équipement utilise les paramètres de configuration. L'équipement est opérationnel. ◦ Si les paramètres de configuration (exécution) **n'existent pas** dans la mémoire Flash, l'équipement utilise les paramètres de configuration par défaut et est opérationnel. **Remarque** : Les paramètres par défaut ne sont pas forcément adaptés à votre système.
l'équipement ne détecte pas d'adresse IP dans la mémoire Flash,	l'équipement utilise l'adresse IP par défaut provenant de l'adresse MAC de l'équipement. • L'équipement est en mode E/S sûr. Les E/S ne sont pas opérationnelles.

Tableau. A.7. Processus d'obtention d'une adresse IP par un Momentum

Puisque dans plusieurs applications, on ne dispose pas de serveur BOOTP/DHCP, la méthode la mieux adaptée est d'enregistrer pour chaque équipement son adresse IP dans la mémoire flash. Cette méthode possède en plus l'avantage d'avoir une maitrise des adresse IP des équipements puisque c'est le concepteur lui-même qui les introduit.

Pour enregistrer une adresse IP dans la mémoire flash de l'adaptateur, il faut suivre les étapes suivantes :

Etape	Action
1	Accédez à la page de l'équipement intitulée Configuration de l'adaptateur de communication.
2	Définissez les paramètres de configuration (exécution) de votre système.
3	Accédez à la page Web Configurer les paramètres IP.
4	Saisissez les paramètres IP.
5	Sélectionnez la case à cocher **Retour aux paramètres IP enregistrés en l'absence d'un serveur d'adresses** (témoin IPFallBack).
6	Saisissez le mot de passe de configuration (**httpcfguser**).
7	Cliquez sur le bouton **Mettre à jour les valeurs enregistrées**.
8	Sélectionnez redémarrer.

Tableau. A.8. Procédure d'enregistrement d'une adresse IP dans la mémoire flash de l'adaptateur de communication.

Au redémarrage, l'adaptateur Momentum 170ENT11001 émet des requêtes BOOTP pendant 30 secondes et utilise ensuite l'adresse IP enregistrée dans la mémoire Flash.

3.6 Accès à l'adaptateur de communication Ethernet 170ENT11001 via les registres internes

L'adaptateur contient trois groupes de registres internes qui lui permettent :

- de transférer des données d'entrée et de sortie aux borniers de l'embase via les registres du groupe de données.
- de définir ou de récupérer la configuration de l'adaptateur via les registres du groupe de configuration.
- de contrôler l'état du système. Via registres du groupe d'état.

Il est possible d'accéder aux trois groupes de registres par les blocs fonction MSTR (fonctions de communication sur un réseau) sur le réseau Ethernet par des commandes Modbus standard. L'utilisateur peut accéder à plusieurs registres pour obtenir des informations sur le module E/S via le réseau Ethernet.

Il est possible d'accéder à ces mêmes registres via les pages Web incorporées dans l'adaptateur de communication. L'accès via ces pages Web permet d'obtenir une interface graphique plus conviviale pour configurer les registres.

Le schéma de la **figure II.11** illustre le flux de données entre les registres internes du réseau Ethernet et de l'adaptateur de communication Ethernet.

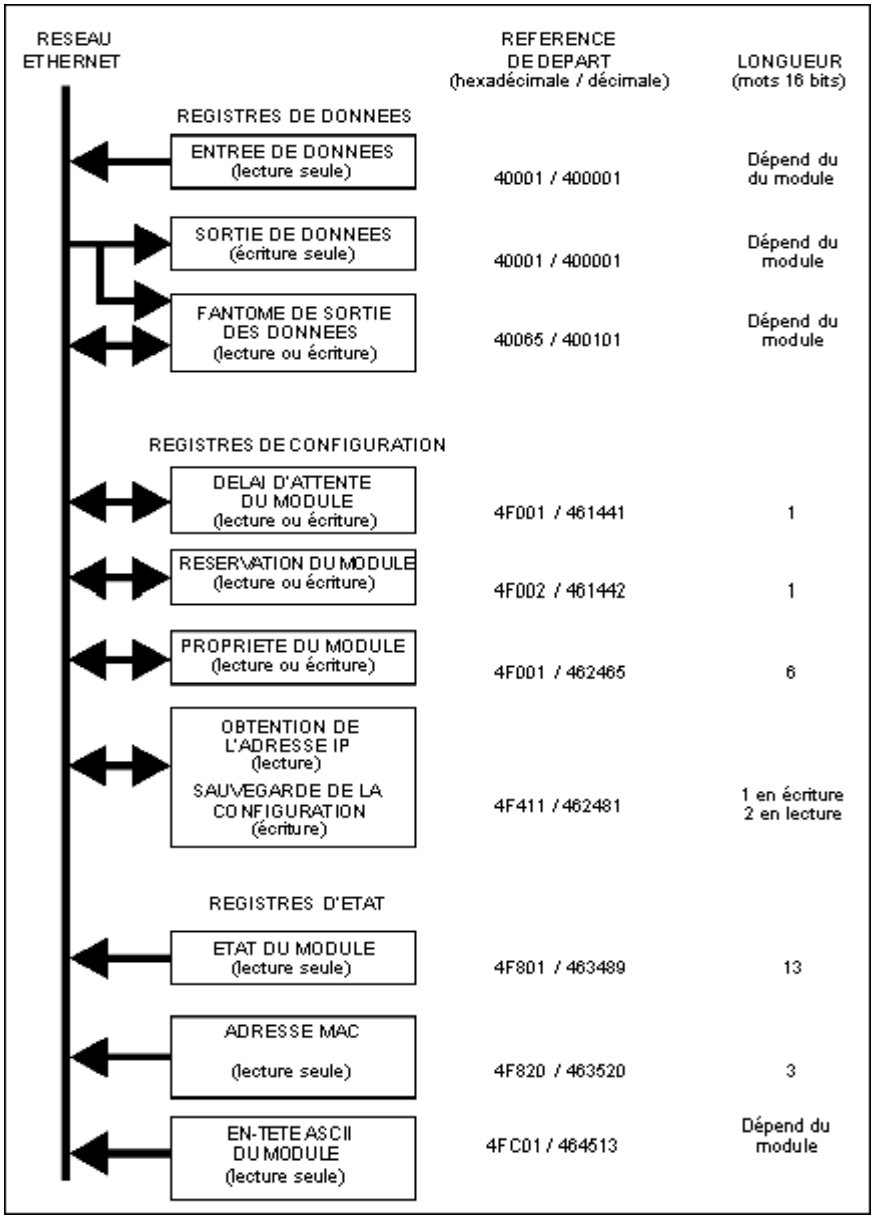


Fig. A.11. Flux de données entre les registres interne de l'adaptateur et le réseau Ethernet

4. Conclusion

Ce chapitre nous a permet d'avoir une vue global sur les différentes gammes de Schneider de Schneider Electric pour les automates programmables, nous avons aussi eu un aperçu sur les modules d'E/S déportés qui permettent d'économiser grandement dans le câblage et se trouvent ainsi très utiles dans le monde de la GTB.

ANNEXE B

Introduction aux Réseaux Informatique

1. Introduction

Dans le monde de la Gestion Technique des Bâtiments, les réseaux informatiques occupent une place primordiale, c'est pourquoi nous leurs consacreront ce chapitre qui donne les notions de base sur les réseaux ce qui va nous permettre de mieux appréhender les chapitres qui suivront.

2. Définition et besoins d'un réseau informatique

Un *réseau informatique* est un ensemble d'ordinateurs ou d'équipements reliés ensemble par des supports de transmission (medias) qui leur permettent d'échanger des données entre eux.[B7]

Les réseaux de données sont apparus à la suite des applications informatiques écrites pour les entreprises. Cependant, au moment où ces applications ont été écrites, les entreprises possédaient des ordinateurs qui étaient des machines autonomes, fonctionnant seules et indépendamment les unes des autres. Très vite, on s'aperçut que cette façon d'exploiter les entreprises n'était ni efficace ni rentable. Les entreprises avaient besoin d'une solution qui apporte des réponses aux trois questions suivantes :

1-Comment éviter la duplication de l'équipement et des ressources ?

2-Comment communiquer efficacement ?

3-Comment mettre en place et gérer un réseau ?

Les entreprises ont pris conscience des sommes qu'elles pouvaient économiser et des gains de productivité qu'elles pouvaient réaliser en utilisant la technologie réseau. Elles ont commencé à ajouter des réseaux et à étendre les réseaux existants presque aussi rapidement que l'apparition des nouvelles technologies et des nouveaux produits de réseau le permettait. Conséquence : au début des années 1980, la technologie des réseaux a connu une croissance phénoménale, mais ce développement était chaotique à plusieurs points de vue.

Vers le milieu des années 80 des problèmes de compatibilité entre réseaux utilisant des technologies (logicielles ou matérielles) différentes sont apparus.

Pour remédier à ces problèmes, une première solution utilisant les réseaux locaux **LAN**(Local Area Network) a été proposée afin de relier les différentes stations de travail, terminaux et autres unités et périphériques d'un immeuble ou d'une entreprise.

Avec le temps même les réseaux LAN étaient devenus insuffisants vu que les entreprises se développaient à l'échelle mondiale. Il fallait donc trouver un moyen de relier ces entreprises, éloignés géographiquement, afin qu'elles échangent plus facilement leurs informations. Les

réseaux WAN (Wide Area Network) sont alors apparus reliant ainsi les réseaux locaux entre eux et leurs ont ainsi donné accès aux ordinateurs et aux serveurs de fichiers situés en d'autres lieux. [W5]

La figure III.1 présente des exemples de réseaux locaux LAN et et de réseaux WAN :

Distance entre les unités centrales	Emplacement des unités centrales	Nom
0,1 m	Circuit imprimé Assistant personnel	Carte-mère Réseau personnel (PAN)
1,0 m	Millimètre Mainframe	Réseau de systèmes informatiques
10 m	Salle	Réseau local (LAN) Votre salle de classe
100 m	Bâtiment	Réseau local (LAN) Votre école
1000 m = 1 km	Campus	Réseau local (LAN) Université de Provence
100 000 m = 100 km	Pays	Réseau longue distance (WAN) Cisco Systems, Inc.
1 000 000 m = 1 000 km	Continent	Réseau longue distance (WAN) Europe
10 000 000 m = 10 000 km	Planète	Réseau longue distance (WAN) Internet
100 000 000 m = 100 000 km	Système terre-lune	Réseau longue distance (WAN) La terre et les satellites artificiels

Fig. B.1 Exemples de réseaux LAN et WAN [W5]

3. Model de référence OSI de l'ISO

Développer des plates-formes logicielles et matérielles pour permettre à des ordinateurs de communiquer est une tache très compliquée même si les éléments de ces plates-formes sont construits par un seul fabriquant. Ces derniers pour augmenter leur rentabilité, tendent à optimiser leurs produits les rendant ainsi plus complexes. Cette complexité croissante des produits conduit souvent à des problèmes de compatibilité.

Le besoin de permettre à des systèmes basés sur des processeurs de communiquer entre eux a conduit l'ISO (l'Organisation Internationale pour la Normalisation) à **proposer un model réseaux standard** connu sous le nom de model **OSI** (Open System Interconnexion). Le but de l'ISO en développant ce Model était d'offrir des standards acceptables par tout les constructeurs, les aidant ainsi à concevoir des produits interopérables et donc plus compétitifs.

La maitrise de la difficulté de la communication réseau peut être effectuée en utilisant une architecture en couche dans laquelle toutes les fonctions réseaux sont partitionnées en divers groupes appelés « **couches** ». Dans ces conditions, les couches supérieures utilisent les services offerts par les couches inférieurs sans se soucier de la manière dont ces couches inferieures traitent leurs informations et produisent ces services.

Le model OSI définit un certain nombre de couches, les fonctions effectuées par chaque couche et les interfaces entre les différentes couches (**passerelles**) [B1].

3.1. Avantages du model de référence OSI [W5]

Le modèle de référence OSI est le principal modèle des communications réseau. Bien qu'il en existe d'autres, la majorité des fournisseurs de solutions réseau relient aujourd'hui leurs produits à ce modèle de référence et l'utilisent pour former les utilisateurs de leurs produits et ceci parce que le modèle de référence OSI permet de voir les fonctions réseau exécutées au niveau de chaque couche. Plus important encore, ce modèle de référence constitue un cadre que les utilisateurs peuvent utiliser pour comprendre comment les informations circulent dans un réseau. Ils peuvent en outre se servir du modèle de référence OSI pour visualiser comment les informations, ou paquets de données, circulent à partir des programmes d'application (ex: tableurs, documents, etc.), en passant par un média réseau (ex. : fils, etc.), jusqu'à un autre programme d'application se trouvant sur un autre ordinateur en réseau, même si l'expéditeur et le destinataire utilisent des types de médias réseau différents.

Le modèle de référence OSI comporte sept couches numérotées, chacune illustrant une fonction réseau bien précise. Cette répartition des fonctions réseau est appelée *organisation en couches*. Le découpage du réseau en sept couches présente les avantages suivants :

- Il permet de diviser les communications sur le réseau en éléments plus petits et plus simples.
- Il uniformise les éléments du réseau afin de permettre le développement et le soutien multi constructeur.
- Il permet à différents types de matériel et de logiciel réseau de communiquer entre eux.
- Il empêche les changements apportés à une couche d'affecter les autres couches, ce qui assure un développement plus rapide.
- Il divise les communications sur le réseau en éléments plus petits, ce qui permet de les comprendre plus facilement.

3.2. Les sept couches du model de référence OSI

Dans le modèle de référence OSI, le problème consistant à déplacer des informations entre des ordinateurs est divisé en sept problèmes plus petits et plus faciles à gérer et chacun des problèmes à résoudre est représenté sous forme d'une couche. Le model de référence OSI se compose de sept couches représentées par la **figure B.2**

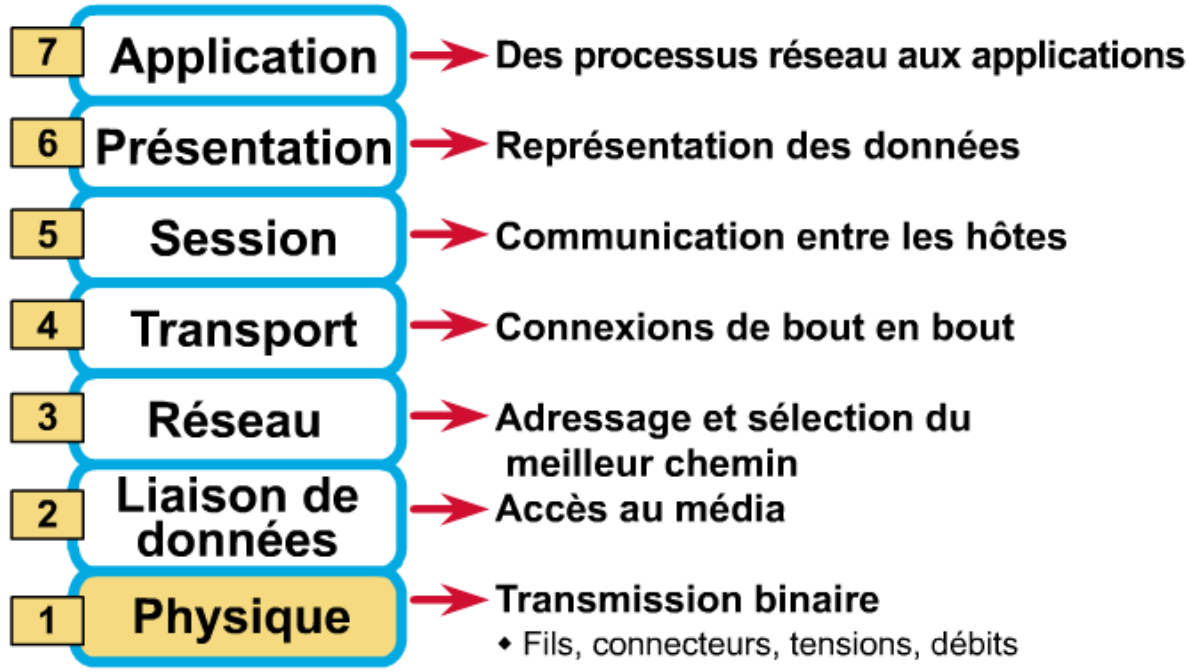


Fig. B.2 Les sept couches du model de référence OSI [W5]

3.2.1. Couche 7 : La couche Application

La couche application est la couche OSI la plus proche de l'utilisateur. Elle fournit des services réseau aux applications de l'utilisateur. Elle se distingue des autres couches en ce sens qu'elle ne fournit pas de services aux autres couches OSI, mais seulement aux applications à l'extérieur du modèle OSI. Voici quelques exemples de ce type d'application : tableurs, traitements de texte et logiciels de terminaux bancaires. La couche application détermine la disponibilité des partenaires de communication voulus, assure la synchronisation et établit une entente sur les procédures de correction d'erreur et de contrôle d'intégrité des données. [B7]

3.2.2. Couche 6 : La couche Présentation

La couche présentation s'assure que les informations envoyées par la couche application d'un système sont lisibles par la couche application d'un autre système. Au besoin, la couche présentation traduit différents formats de représentation des données en utilisant un format commun. [B7]

3.2.3. Couche 5 : La couche Session

La couche session est chargée d'établir une connexion entre les ordinateurs communiquant. Pour se faire, cette couche s'assure que le destinataire est bien présent.

Une fois cette connexion (ou session) établie, cette couche s'assure que le dialogue se déroule correctement et que la session s'achève convenablement. [B7]

3.2.4. Couche 4 : La couche Transport

Cette couche reçoit les données de la couche session et les découpe en fragments de données, plus facile à transporter, appelés **paquets**. Cette couche transmet ensuite tous les paquets à la couche réseau et s'assure qu'ils arrivent à destination.

A la réception des données, la couche transport assemble les paquets pour former le message originel et envoie un « un accusé de réception » pour indiquer que les paquets sont arrivés correctement.

Dans le cas ou les paquets arrivent de manière désordonnée (communication sans connexion), la couche transport reconstitue le message (elle l'ordonne à condition que la couche transport de l'ordinateur expéditeur numérote les paquets).

La couche transport s'occupe aussi du **contrôle de flux**, c.à.d. qu'elle s'assure l'émetteur n'envoie pas de données plus vite que le récepteur ne peut les recevoir. [B7]

3.2.5. Couche 3 : La couche réseaux

Cette couche est chargée de l'adressage du réseau et d'indiquer aux paquets les directions à suivre pour arriver à destination. [B7]

3.2.6. Couche 2 : La couche liaison de données

Cette couche organise les paquets de données issus de la couche transport en **trames** car un paquet ne peut être directement transmis sur un media. En effet on ne peut distinguer les débuts des paquets ni leurs fin.

Une **trame** est l'entité transportée sur le support de transmission, c'est en fait un paquet dont on peut reconnaitre le début et la fin.

La couche liaison de données assure aussi un transit fiable des données sur une liaison physique. Ainsi, la couche liaison de données s'occupe de la notification des erreurs, de la livraison ordonnée des trames.

Elle définit aussi des règles pour le partage du réseau par les différents nœuds **(équipements du système)**. Dans un réseau LAN qui a une topologie BUS, la couche physique est partagée suivant une division temporelle. L'information est transférée sous forme de trames adressées, une trame à la fois. Le format de ces trames et les méthodes par lesquelles un nœud décide de recevoir ou d'émettre ces trames est également définit dans la couche 2. [B7]

3.2.7. Couche 1 : La couche physique

La couche physique définit les spécifications électriques (de même que radio et optique), mécaniques, procédurales et fonctionnelles permettant d'activer, de maintenir et de désactiver la liaison physique entre les systèmes. Les caractéristiques telles que les niveaux de tension, la synchronisation des changements de tension, les débits physiques, les distances maximales de transmission, les connecteurs physiques et d'autres attributs semblables sont définies par la couche physique. Cette couche s'occupe donc des signaux et médias. [W5]

4. Model de référence TCP/IP [W5]

Le TCP/IP (Transmission Control Protocol/Internet Protocol) a été crée par le ministère américain de la décence parce qu'il avait besoin d'un réseau capable de résister à toutes les conditions, même à une guerre nucléaire. Ce protocole est la norme de référence sur laquelle repose Internet. Le protocole TCP/IP *IP* rend possible l'échange de données entre deux ordinateurs, partout dans le monde, à une vitesse quasi équivalente à celle de la lumière.

Le modèle TCP/IP comporte quatre couches : la couche application, la couche transport, la couche Internet et la couche d'accès au réseau. Comme vous pouvez le constater, certaines couches du modèle TCP/IP portent le même nom que des couches du modèle OSI. Il ne faut pas confondre les couches des deux modèles, car la couche application comporte des fonctions différentes dans chaque modèle.

4.1. Les quatre couches du model de référence TCP/IP

La **figure B.3** montre les quatre couches du mode TCP/IP

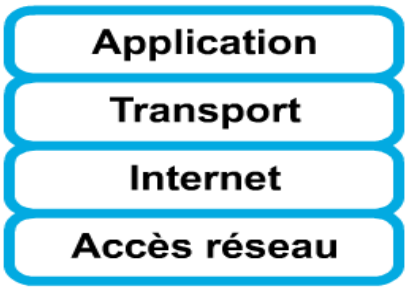


Fig. B.3 Les couches du model TCP/IP [W5]

4.1.1. Couche 1 : La couche Application

Les concepteurs du modèle TCP/IP estimaient que les protocoles de niveau supérieur devaient inclure les détails des couches session et présentation. Ils ont donc simplement créé une couche application qui gère les protocoles de haut niveau, les questions de représentation, le code et le contrôle du dialogue. Le modèle TCP/IP regroupe en une seule couche tous les aspects liés aux applications et suppose que les données sont préparées de manière adéquate pour la couche suivante. [W5]

4.1.2. Couche 2 : La couche Transport

Cette couche est chargée d'assurer la fiabilité du transport de donnée grâce à un contrôle de flux et à une correction des erreurs. L'un des protocoles de cette couche est le TCP (Transmission Control Protocol - protocole de contrôle de transmission) qui assure une communication fiable présentant un taux d'erreurs peu élevé. . Le protocole TCP est orienté connexion c.à.d. qu'il établit un dialogue entre l'ordinateur source et l'ordinateur de destination pendant qu'il prépare les informations de couche application en unités appelées segments. [W5]

4.1.3. Couche 3 : La couche Internet

Cette couche a pour rôle d'envoyer les paquets source d'un réseau quelquonque vers sa destination et ceci indépendamment des réseaux à traverser pour y parvenir et ceci en utilisant le protocole IP (Internet Protocol). En d'autre terme, ce protocole identifie le meilleur chemin possible pour la circulation des données. [W5]

4.1.4. Couche 4 : La couche d'accès au réseau

Cette couche se charge de tout ce dont un paquet IP a besoin pour établir une liaison physique, puis une autre liaison physique. Cela comprend les détails sur les technologies LAN et WAN, ainsi que tous les détails dans les couches physiques et liaison de données du modèle OSI. [W5]

4.2. Les différents protocoles utilisés par le model de référence TCP/IP

Chaque couche du model de référence TCP/IP peut utiliser un ou plusieurs protocoles comme l'indique le schéma suivant :

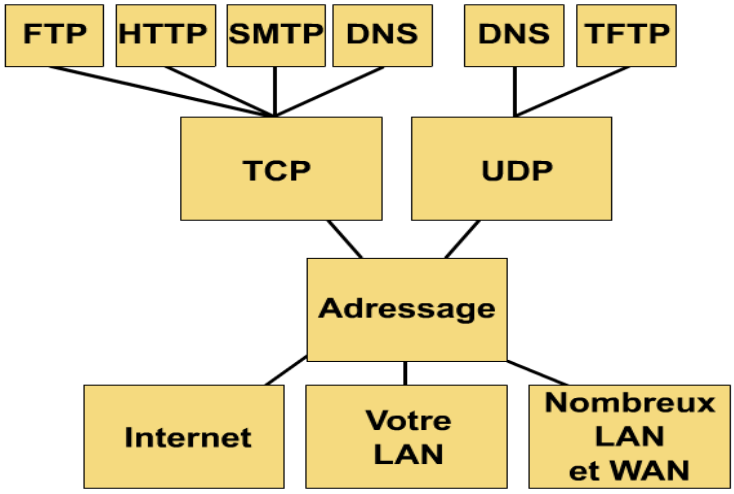


Fig. B.4 Schéma de protocoles TCP/IP [W5]

Les protocoles de la couche application sont donc :

- FTP - Protocole de transfert de fichiers ou protocole FTP
- HTTP - Protocole HTTP (HyperText Transfer Protocol)
- SMTP - Protocole SMTP (Simple Mail Transfer Protocol)
- DNS - Système DNS (Domain Name System)
- TFTP - Protocol TFTP (Trivial File Transfer Protocol)

Dans le modèle TCP/IP, IP (Internet Protocol) est le seul et unique protocole utilisé dans la couche Internet, et ce, quels que soient le protocole de transport utilisé et l'application qui demande des services réseau. La raison est que IP est un protocole universel qui permet à tout ordinateur de communiquer en tout temps et en tout lieu.

5. Les réseaux LAN (Local Area Network) [B7]

Les réseaux locaux sont mis en œuvre pour atteindre Les principaux objectifs suivants :

- **Le partage des ressources physiques :** permet de partager des périphériques entre plusieurs ordinateurs. Un exemple typique est l'imprimante qui peut être commune à un ensemble de postes de travail.
- **Le partage d'applications :** En effet, il s'avère quelquefois très intéressant d'installer des applications utilisées par beaucoup d'ordinateurs sur un seul disque partagé en réseau plutôt que de l'installer sur tous les ordinateurs. On peut par exemple installer un logiciel qui demande une forte puissance de calcul sur un ordinateur puissant et partager ce logiciel par un ensemble de stations de travail peu performantes.
- **Le partage de données :** Dans une entreprise, plusieurs fichiers peuvent être partagés entre les salariés directement sur le réseau limitant ainsi les déplacements inutiles.
- **La communication entre personnes :** Un exemple typique de cette application est l'e-mail.

6. Topologies des réseaux LAN [B7]

La topologie du réseau représente la façon d'interconnecter les ordinateurs et équipements entre eux. A chaque topologie ses avantages et ses inconvénients en terme de fiabilité, complexité et du cout. On distingue deux définitions de la topologie :

- **La topologie physique :** vue par l'utilisateur en suivant le câblage.
- **La topologie logique :** La façon avec laquelle les données circulent sur le réseau.

6.1. Topologie en Bus

Du point de vue physique, cette topologie utilise un seul câble appelé **bus** pour relier les ordinateurs entre eux. Le bus doit être terminé sur ces deux extrémités par un anti-réflecteur de signal appelé **bouchon**.

Du point de vue logique, un seul ordinateur à la fois peut émettre des données qui circuleront tout au long du bus et recueillis par l'ordinateur destinataire.

Si un ordinateur tombe en panne, cela ne perturbe pas le réseau en revanche si le câble se rompt ceci conduit à la défaillance du réseau, de plus la longueur du bus est limitée à cause de l'affaiblissement du signal.

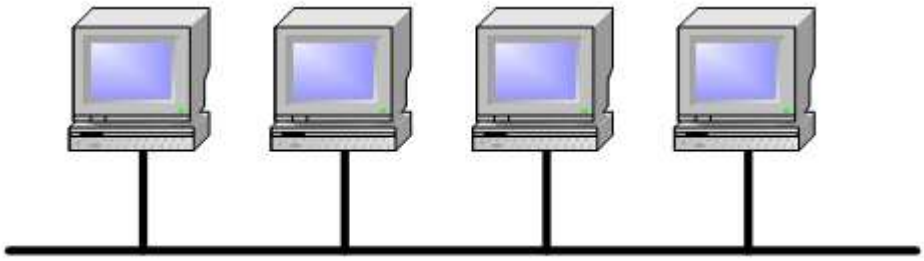


Fig. B.5 Topologie en Bus

6.2. Topologie étoile

Du point de vue physique, tout les ordinateurs sont reliés à un élément central appelé **concentrateur**.

Du point de vue logique, quand un ordinateur envoie des données au concentrateur, ce dernier peut les renvoyer soit à tous les autres ordinateurs soit seulement au destinataire.

Si un des ordinateurs du réseau tombe en panne, le réseau n'est pas influencé, cependant si le concentrateur tombe en panne alors il survient une défaillance complète du réseau.

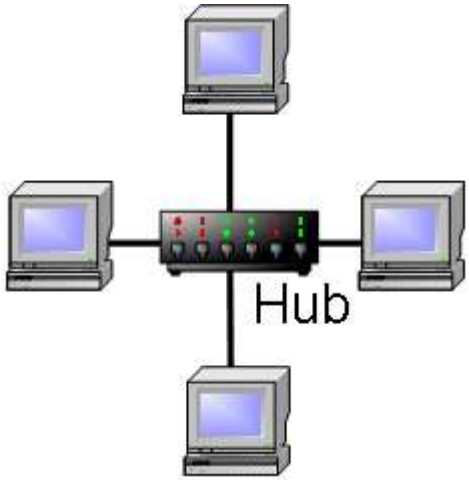


Fig. B.6 Topologie en Etoile

6.3. Topologie anneau

Du point de vue physique, un anneau de câble permet de relier les ordinateurs entre eux.

Du point de vue logique, les données circulent d'un ordinateur à un autre jusqu'à arriver au destinataire.

On voit bien que s'il y a une panne d'un ordinateur ou du câble, alors le réseau entier est perturbé.

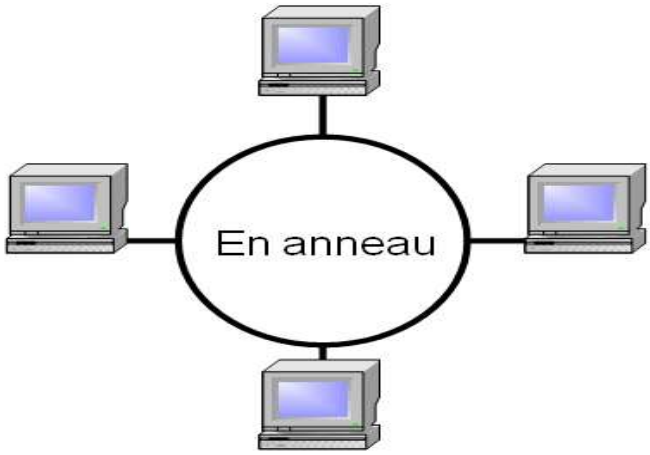


Fig. B.7 Topologie en anneau

7. Méthodes d'accès au Média [B7]

La problématique majeur d'accès au média est de savoir quel ordinateur accède au support de transmission, à quel moment et pour combien de temps.

Une méthode d'accès au support de transmission (Medium Access Control en anglais) représente l'ensemble des règles qui définissent la façon dont un ordinateur place les données sur le support et récupère les données provenant de ce dernier. On distingue deux types de méthodes d'accès au support de transmission :

7.1. Méthodes Statiques

Dans cette méthode, on alloue à chaque ordinateur un intervalle de temps dans lequel il peut émettre et recevoir. L'inconvénient de cette méthode est que si un ordinateur n'a rien à émettre ni à recevoir, une possibilité d'exploitation du support sera perdue.

7.2. Méthodes Dynamiques

Dans cette méthode il n'y a pas d'allocation du support car ce dernier n'est utilisé que par les ordinations qui en ont besoin. C'est cette méthode qui est utilisée dans les réseaux LAN. On distingue dans cette méthode :

7.2.1. La gestion Centralisée : Dans ce cas, un ordinateur est responsable de l'accès au support.

7.2.2. La gestion décentralisée : Il n'y a pas d'ordinateurs maitre ou esclave, on distingue :

- **L'accès aléatoire :** L'ordinateur émetteur, émet ses données sur le canal dés qu'il n'y a aucune activité sur ce dernier.
- **L'accès contrôlé par un droit de parole :** Ce droit est matérialisé par un **jeton** (*Token* en anglais) et seul l'ordinateur possédant ce jeton a le droit d'émettre ou de

recevoir des données. A la fin de la transmission, l'émetteur passe le jeton à l'ordinateur suivant.

8. Normalisation des réseaux LAN [B7]

Les deux principaux organismes ayant contribué aux travaux de normalisation sont l'IEEE (Institute of Electrical and Electronics Engineers) aux USA et l'ECMA (Européen Computer Manufactures Association) en Europe.

L'IEEE a crée au cours de l'année 1980 un comité appelé **comité 802** chargé de la normalisation des réseaux locaux. L'ECMA créa à son tour un comité l'année suivante. L'objectif de ces deux organismes était la normalisation des réseaux locaux pour assurer l'interopérabilité entre les équipements de divers fabricants. Ces normes sont connus sous le nom de **IEEE 802** qui ont été adoptés par l'ISO pour devenir la norme **ISO 8802**.

La norme **IEEE 802** se limite se limite aux couches liaison de donnée (couche 2) et physique (couche 1) du model OSI, de plus cette norme divise la couche liaison en deux sous couches :

- Contrôle des liaisons logiques ou LLC (Logical Link Control) : cette sous couche établit les liaisons, ordonne les trames, envoie les accusés de réception et contrôle le trafic.
- Contrôle d'accès au support ou MAC (Media Access Control) : cette sous couche gère l'accès au média (support) en garantissant qu'un seul ordinateur ou équipement est en train de transmettre à la fois. Elle organise aussi les données en trames et détecte les erreurs de transmission.

Les principales normes IEEE 802

- **IEEE 802.3 :** réseau Ethernet (topologie logique en bus et méthode d'accès CSMA/CD)
- **IEEE 802.4 :** réseau Token Bus (topologie logique en bus avec méthode d'accès à jeton)
- **IEEE 802.5 :** réseau Token Ring (topologie logique en anneau avec méthode d'accès à jeton)
- **IEEE 802.11 :** réseaux locaux sans fil.

Les normes les plus utilisés dans les réseaux locaux sont celles qui concernent les réseaux Ethernet (IEEE 802.3) et Token Ring (IEEE 802.5)

9. Réseau Ethernet (IEEE 802.3)

Les réseaux Ethernet sont les plus répandus dans le monde, en effet plus de 90% des réseaux locaux mondiaux sont de ce type car ils sont simple à mettre en œuvre et leurs cout est très faible.

Les principales caractéristiques de ces réseaux sont illustrées par le tableau suivant :

CARACTERISTIQUES	**DESCRIPTION**
Topologie Logique	Bus
Mode de transmission	Mode de base
Méthode d'Accès au support	CSMA/CD
Spécifications	IEEE 802.3
Débit binaire	10 Mbps à 1 Gbps
Support de transmission	Câble coaxial, paire torsadée ou fibre optique
Longueur de la trame	Entre 64 à 1518 octets

Tableau. B.1 Topologie en anneau

9.1. Méthode d'accès CSMA/CD

La méthode d'accès CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) veut dire : Accès Multiple avec écoute de la porteuse et détection de collisions.

C'est une méthode dynamique à gestion décentralisée et accès aléatoire fonctionnant dans la sous couche MAC.

Cette méthode fonctionne selon le principe suivant :

Lorsqu'un ordinateur veut émettre des données sur le canal de transmission, il commence par écouter ce dernier pour détecter un éventuel signal émis par d'autres ordinateurs (Carrier Sense). Si le canal est libre, l'ordinateur envoie sa trame qui sera *diffusée* tout au long du canal de transmission et qui sera captée par le destinataire.

Tout les ordinateurs de ce réseau peuvent émettre de la même manière sur le média c'est pourquoi on parle d'accès multiple (Multiple Access).

Si deux ou plusieurs ordinateurs émettent simultanément (car ils ont détecté que le canal était libre au même moment), les signaux émis se superposent et deviennent inexploitable, on dit que les signaux entrent en **collision**. Tout les ordinateurs sur le réseau doivent pouvoir détecter la collision en particulier les ordinateurs qui sont à l'origine de cette collision.

Lors d'une collision, les ordinateurs à l'origine arrêtent immédiatement d'émettre et génèrent un signal avertissant les autres stations de la collision puis ils attendent **un temps aléatoire** avant de réémettre leur données. Ce temps aléatoire est généré par un algorithme et assure que le temps de réémission des données est différent pour chacune des stations émettrices. [B7]

La **figure B.8** illustre le principe de la méthode CSMA/CD :

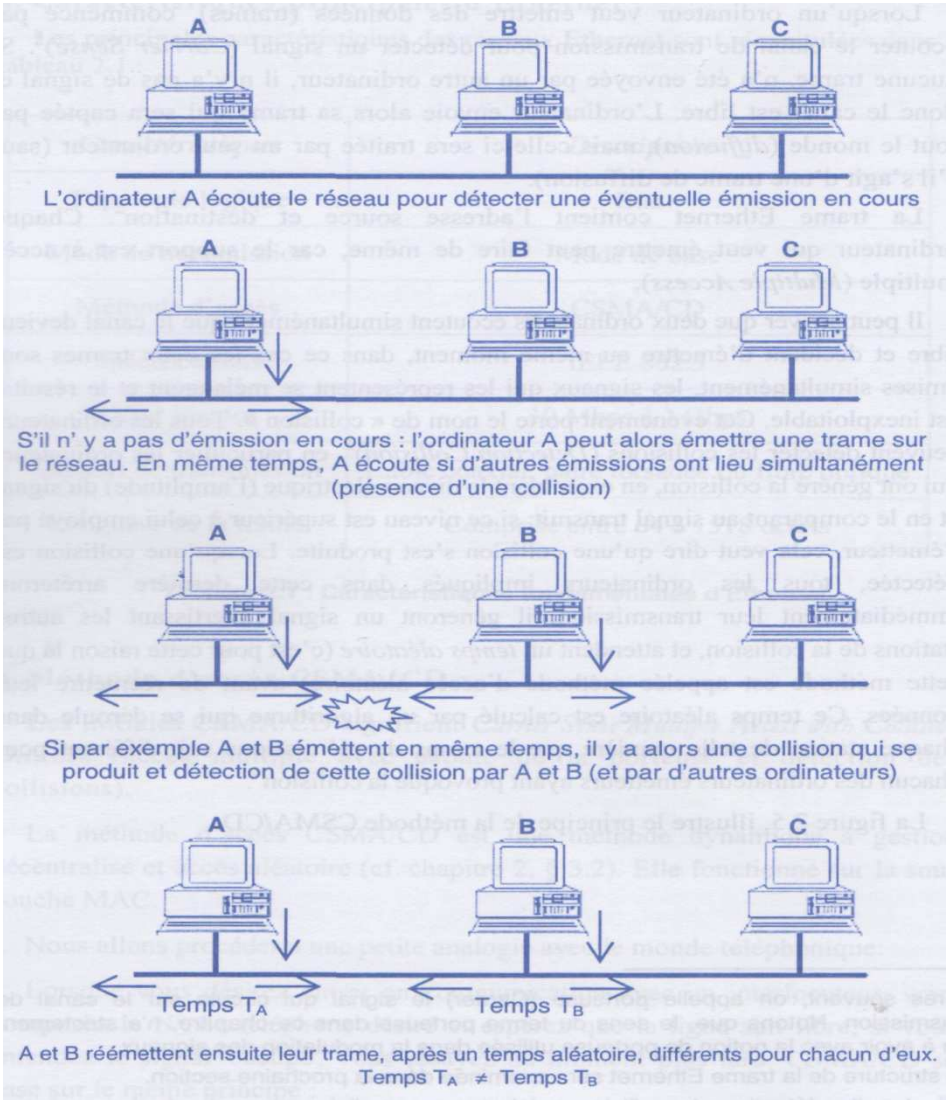


Fig. B.8 Principe de la méthode CSMA/CD [B7]

9.2. Format de la trame Ethernet

La structure d'une trame Ethernet est illustrée par la **figure B.9**

PRE	SFD	DA	SA	LEN	DATA	PAD	FCS
7 octets	1 octet	6 octets	6 octets	2 octets	0-1500 octets	0-46 octets	4 octets

Fig. B.9 Structure de la trame Ethernet [B7]

- PRE (Préambule) : une suite de 10101010 est envoyée 7 fois permettant aux horloges des récepteurs de se synchroniser à celles de l'émetteur.

- SFD (Start Frame Delimiter) **délimiteur de trame** : il contient la séquence 10101011 qui permet de délimiter le début de la trame.
- DA (Destination Adress) **adresse de destination** : Contient l'adresse MAC de l'équipement destinataire.
- SA (Source Adress) **adresse de la source** : Contient l'adresse MAC de la source
- LEN (LENgth) **longueur** : Ce champ précis la longueur de la donnée DATA en octets.
- DATA : Ce champ contient la donnée effective (utile) à transmettre sur le réseau.
- PAD (PADding) **Bourrage** : la norme IEEE 802.3 stipule que la longueur minimum d'une trame valide doit comporter 64 octets du champ DA jusqu'au champ FCS donc si le champ DATA est inférieur à 46 octets, la trame doit être complétée par des bits de remplissage dans le champ PAD pour atteindre la taille minimum spécifiée par le protocole.
- FCS (Frame Check Sum) : ce champ sert à effectuer la détection d'erreurs de transmission. [B7]

9.3. Adressage dans les réseaux Ethernet [B7]

Dans un réseau local, tout ordinateur ou équipement dispose d'une adresse *physique* appelée adresse **MAC** (car c'est la sous couche MAC qui s'occupe de cet adressage) stockée en mémoire sur la carte réseau.

Ces adresses MAC sont administrées par l'IEEE, en effet chaque carte réseau est affectée d'un numéro unique au niveau mondial pour éviter que différentes cartes même venant de différents constructeurs ait la même adresse MAC.

Le format d'une adresse Ethernet est le suivant :

48 bits			
I/G	U/L	Identification par l'EEEE du Constructeur	Numéro de série attribué par le Constructeur
1 bit	1 bit	22 bits 2^{22} constructeurs possibles	24 bits

Tableau. B.2 Topologie en anneau

Un identifiant unique est attribué à tout fabriquant de matériel réseau par l'IEEE et le numéro de série est attribué par le fabriquant lui-même en garantissant son unicité.

- Si le bit I/G (Individual/Group) de l'adresse Ethernet vaut 0 (I/G=0) l'adresse est dite **unicast** et elle est utilisée pour adresser un ordinateur dans le réseau.
- Si I/G=1 l'adresse est dite **multicast** et elle est utilisée pour adresser un groupe d'ordinateurs dans le réseau.

- Si tous les bits de l'adresse Ethernet sont à 1 (FFFFFFFFFFFF) l'adresse est dite **broadcast** et sert à adresser tout les ordinateurs du réseau.
- Si U/L=0 alors l'adresse est conforme aux normes IEEEE.
- Si U/L=1 c'est l'utilisateur qui décide des adresses à affecter aux cartes réseau.

9.4. Les différents types de réseaux Ethernet [B7]

Il existe plusieurs types de réseaux Ethernet se différenciant principalement au niveau de la couche physique (topologie physique utilisée, type et longueur du câble, les types de connecteurs….). Les 3 principales familles de l'Ethernet sont les suivantes :

9.4.1. Ethernet 10 Mbps

Il éxiste plusieurs spécifications pour la couche physique de l'Ethernet 10 Mbps conduisant conduisant à l'éxistance de plusieurs types normalisés de cette famille selon IEEE sous la forme **10BaseX** avec :

- 10 : correspond au débit maximal de 10 Mbps.
- Base : indique le réseau transmet les données suivant la technique de Bande de Base.
- X :indique la distance maximale en centaine de métres du cable si X est un nombre, en revanche si X est une lettre, il désigne le type de cable utilisé.

Ainsi on trouve pour l'Ethernet 10 Mbps :

- **10Base5(ou Thick Ethernet) :** qui est aussi connu sous le nom de **standard Ethernet**.Comme son nom l'indique, il utilise un cable coaxial épais qui peut aller jusqu'à 500m et transportant des données à un débit de 10Mbps. Ce type de réseau est basé sur la topologie physique en bus.
- **10Base2 (Thin Ethernet) :**Il est réalisé avec un cable coaxial fin pouvant aller jusqu'à 200m et ayant un débit de 10 Mbps. Il est plus économique et plus facil à istaller que le précédent.
- **10BaseT :**Ce type de réseau Ethernet repose sur la topologie en étoile et a aujourd'hui détroné les réseaux 10Base5 et 10Base2. Le « T » signifie **Twisted pair cable** ce qui veut dire **cable à paires torsadées**. Ce cable est non blindé, il est plus maniable et moins couteux que le cable coaxial. Il peut aller jusqu'à 100m avec un débit de 10 Mbps.
- **10BaseF :** le « F » ici veut dire **Fibre optique**. Ce réseau est basé sur une topologie en étoile. La distance entre les concentrateurs peut aller pour ce réseau jusqu'à 2Km.

9.4.2. Ethernet 100 Mbps (Fast Ethernet)

Fast Ethernet ne constitue pas une nouvelle norme mais simplement une évolution d'Ethernet 10 Mbps. On distingue :

- **100BaseTX :** qui utilise un câble à paires torsadées non blindé à deux paires catégories 5. La longueur maximale du câble est de 100m et la topologie utilisée est celle en étoile.
- **100BaseT4 :** qui utilise un câble à paires torsadées non blindés à 4 paires de catégorie 3 à 5 et pouvant aller jusqu'à 100m sur une topologie en étoile.
- **100Base FX :** qui utilise des fibres optiques et qui peut être utilisé sur une distance de 2Km

9.4.3. Ethernet 1000 Mbps (Gigabit Ethernet)

C'est la dernière évolution du standard Ethernet pour atteindre des débits de 1 Gbps. On distingue :

- **1000BaseCX :** qui utilise le câble coaxial sur une distance de 25m.
- **1000BaseSX et 1000BaseTX :** qui utilisent tous deux la fibre optique et qui se différencient par le type de cette fibre.
- **1000BaseT :** qui utilise un câble à paires torsadées catégorie 5.

10. Les Unités LAN de base [W5]

10.1. Les cartes réseau (NIC)

Les cartes réseau sont des plaquettes de circuits imprimés logées dans un emplacement d'extension sur la carte-mère d'un ordinateur ou sur un périphérique. Ces cartes, qui contrôlent l'accès de l'hôte au média, agissent au niveau de la couche 2 (liaison de données) du model OSI.



Fig. B.10 Exemple d'une carte réseau [W5]

Comme nous l'avons dit précédemment, chaque carte réseau comporte un identifiant mondialement.

10.2. Les médias

Il existe plusieurs types de médias utilisés dans les réseaux LAN. Nous allons présenter les plus connus d'entre eux :

10.2.1. Câbles à paires torsadées blindées (STP : Shielded Twisted Pair)

C'est un câble de 150Ω dont chaque paire de fils est enveloppée dans une feuille métallique et les quatre paires sont elles-mêmes enveloppées dans une tresse ou une feuille métallique.

Ce câble a l'avantage de réduire l'effet des interférences entre les fils conducteurs (couplage de paire à paire ou diaphonie) ainsi que les bruits venant de l'extérieur du câble (interférences électromagnétiques et interférences de radiofréquences)

Le débit dans ce genre de câbles varie de 10 à 100 Mbps et Il est représenté à la **figure B.11**

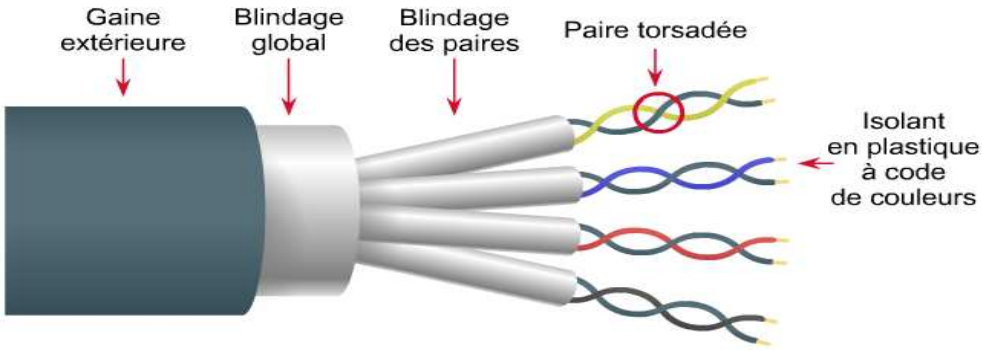


Fig. B.11 Câble à paires torsadées blindées [W5]

10.2.2. Câbles à paires torsadées non blindées (UTP : Unshielded Twisted Pair)

Contrairement au précédent, ce câble ne comporte de blindage, de ce fait il compte uniquement sur l'effet d'annulation produit par les paires torsadées pour limiter la dégradation du signal due aux interférences électromagnétiques.

Un câble à paires torsadées non blindées offre de nombreux avantages. Il est facile à installer et moins onéreux que les autres types de média réseau. Outre le fait qu'il coûte moins cher au mètre que les autres types de câbles LAN, sa taille constitue son principal avantage. Grâce à son diamètre de petite taille, il est possible de loger dans un conduit plus de câbles à paires torsadées non blindées que d'autres types de câble.

Cependant, un câble à paires torsadées non blindées présente aussi des inconvénients. Il est plus sensible au bruit électrique et aux interférences que les autres types de média réseau.

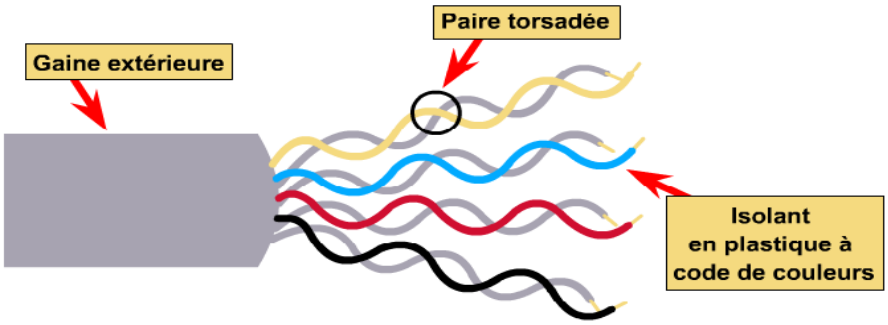


Fig. B.12 Câble à paires torsadées non blindées [W5]

10.2.3. Câble Coaxial

Un *câble coaxial* est un conducteur cylindrique dont le centre est un conducteur en cuivre qui est enveloppé d'un isolant flexible. Sur ce matériau isolant, une torsade de cuivre ou un film métallique protège le conducteur intérieur. Cette deuxième couche, ou blindage, contribue à réduire les interférences externes. La gaine du câble enveloppe ce blindage.

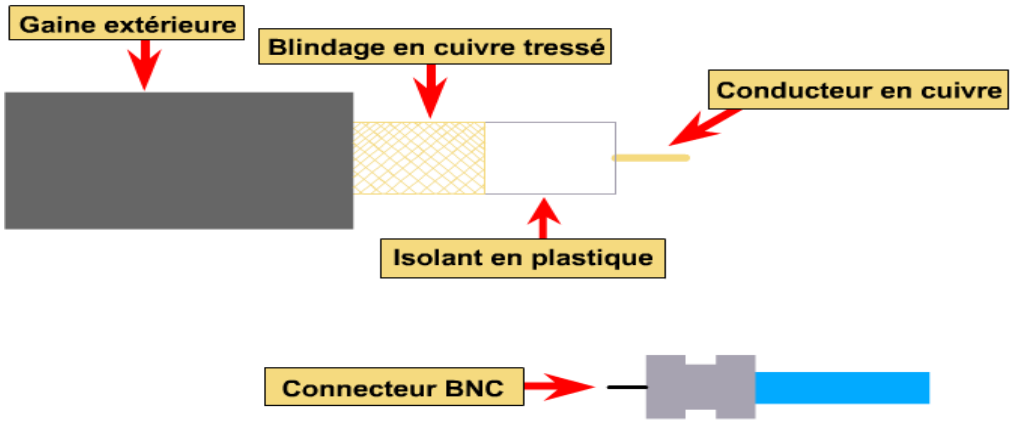


Fig. B.13 Câble coaxial [W5]

Un câble coaxial offre plusieurs avantages pour les réseaux locaux. Il peut couvrir de plus grandes distances entre des nœuds réseau que le câble à paires torsadées blindées ou non blindées, avec moins de répéteurs.

10.2.4. Câble à fibre optique

Un *câble à fibre optique* est un média réseau capable de conduire des impulsions lumineuses modulées. La mise en place d'un réseau à fibre optique revient plus cher que les réseaux avec

des câbles électriques mais la fibre optique est insensible aux interférences électromagnétiques et prend en charge des débits de données considérablement plus élevés.

Le câble à fibre optique utilisé dans les réseaux comprend (**figure III.14**) deux fibres logées dans des enveloppes distinctes. En examinant la coupe transversale d'un câble à fibre optique, on voit que chaque fibre est entourée d'un tampon protecteur généralement en matière plastique tel que le Kevlar et que l'ensemble des fibres est protégé par une gaine externe. Cette gaine protège l'ensemble du câble. La gaine, habituellement en matière plastique, est conforme aux codes du bâtiment et de prévention des incendies. Le but du Kevlar est de fournir une protection supplémentaire aux fibres de verre qui sont très fragiles.

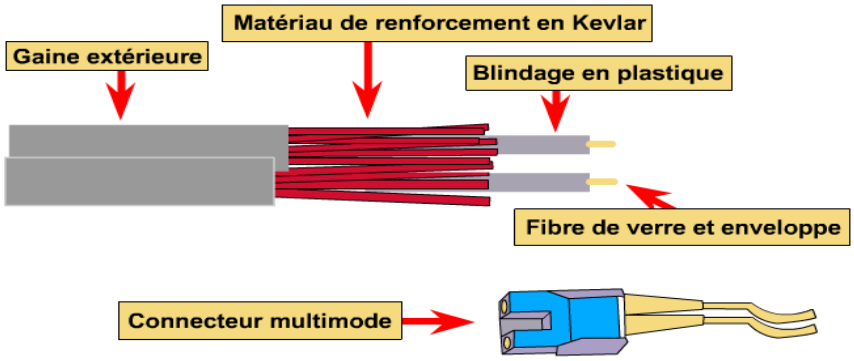


Fig. B.14 Fibre optique [W5]

Les parties de la fibre optique guidant la lumière sont appelées *cœur* et *enveloppe*. Le cœur est habituellement en verre pur dont l'indice de *réfraction* est élevé. Il est enrobé d'une enveloppe de verre ou de matière plastique à faible indice de réfraction afin de retenir la lumière au cœur de la fibre. Ce processus, appelé *réflexion interne totale*, permet à la fibre optique d'agir comme un tuyau guidant la lumière sur de grandes distances, y compris dans les courbes.

10.3. Les répéteurs

L'un des désavantages du câble le plus répandu dans les réseaux LAN (le câble à paire torsadée non blindée de catégorie 5) est la longueur de câble qui est limitée à 100m. Pour prolonger un réseau au-delà de cette limite, nous devons y ajouter une unité appelée *répéteur (ou Hub)*.

Le but du répéteur est de régénérer les signaux réseau et de les resynchroniser au niveau du bit pour leur permettre de voyager sur de plus longues distances dans le média.

Les répéteurs présentent l'inconvénient de ne pas pouvoir filtrer le trafic réseau. Les données (bits) arrivant à un port d'un répéteur sont envoyées à tous les autres ports. Elles sont acheminées vers tous les autres segments LAN, que ce soit nécessaire ou non.

10.4. Les concentrateurs

Le fonctionnement d'un concentrateur est analogue à celui d'un répéteur mais il utilise plusieurs ports de sortie (répéteur multiport). On utilise un concentrateur pour créer un point de connexion central pour le média de câblage.



Fig. B.15 Concentrateur [W5]

10.5. Les ponts

Un pont est une unité de couche 2 conçue pour connecter deux segments LAN. Le rôle du pont est de filtrer le trafic sur un LAN pour conserver le trafic local au niveau local, tout en établissant une connectivité avec d'autres parties (segments) du LAN pour le trafic qui y est destiné. Pour ce faire, un pont utilise les adresses MAC des nœuds (équipements, ordinateurs…) situé de chaque coté du pond et décide quelles seront les trames qui passeront le pont et celles qui resteront au niveau du segment.

Habituellement, on trouve des ponts à deux ports (entrée et sortie) mais qui peut se généraliser à un pont à plusieurs ports (les commutateurs).

10.6. Les commutateurs

Le commutateur n'est rien d'autre qu'un pont multiport. Il ressemble fortement au concentrateur sauf que le commutateur prend des décisions en fonction des adresses MAC tandis que le concentrateur ne prend aucune décision. En raison des décisions qu'il prend, le commutateur rend le LAN beaucoup plus efficace car il commute les données uniquement au port auquel le bon hôte est connecté (l'hôte pour lequel la donnée est destinée). Par contraste, un concentrateur achemine les données à tous les ports, de sorte que tous les hôtes doivent examiner et traiter (accepter ou rejeter) toutes les données.



Fig. B.16 Commutateur [W5]

10.7. Les routeurs

Les routeurs sont des éléments qui travaillent au niveau de la couche 3 (couche réseau) du model OSI ce qui les rend très intéressant du fait qu'ils peuvent relier plusieurs types de réseaux qui peuvent différer de part leurs 2 premières couches (relier par exemple un réseau Ethernet et un réseau Token ring). Du fait que ce sont des éléments de la couche 3, les routeurs prennent leurs décisions en fonction des groupes (classes) d'adresse (par opposition aux ponts et commutateurs qui se basent sur des adresses MAC individuelles).

Le rôle du routeur consiste à acquérir les paquets entrants, de les analyser puis à choisir le meilleur chemin pour les transporter sur le réseau et à les commuter ensuite au port de sortie approprié.

11. Conclusion

Durant ce chapitre, nous avons pu donné les principales notions concernant les réseaux informatiques. C'est sur ces notions que sont élaborés les principaux réseaux dédiés à la GTB que nous verront par la suite.

ANNEXE C

Les Principaux Réseaux Utilisés en GTB

Introduction

L'utilisation de réseaux informatiques dans la Gestion Technique des Bâtiments permet une décentralisation de la régulation augmentant ainsi la fiabilité du système, de plus, le cout du câblage se voit nettement diminué.

Durant ce chapitre, nous allons présenter les deux principaux réseaux utilisés dans le monde de la GTB qui sont le : le réseau Modbus (couche terrain), le réseau LON et le BACnet.

I. Le réseau Modbus

1. Introduction

Le protocole Modbus (marque déposée par Modicon) est un protocole de dialogue basé sur une structure hiérarchisée entre un **maître et plusieurs esclaves**. Il est constitué de trames contenant **l'adresse** de l'automate concerné, la **fonction** à traiter (écriture, lecture), la **donnée** et le **code de vérification d'erreur** appelé contrôle de redondance cyclique sur 16 bits ou CRC16.

La gestion des échanges se fait de la manière suivante :

- Un seul appareil peut émettre sur le bus.
- Les échanges sont gérés par le maître. Seul le maître peut générer des échanges. Aucun esclave ne peut de lui-même envoyer de message sans y avoir été invité
- Le maître réitère la requête lors d'un échange erroné et décrète l'esclave interrogé absent après une non-réponse dans un temps donné.
- L'esclave émet une réponse d'exception au maître s'il ne comprend pas le message ou s'il ne peut pas le traiter. Dans ce cas, le maître peut réitérer ou non la requête.

2. Adressage

Les abonnés du bus sont identifiés par des adresses attribuées par l'utilisateur (Un numéro de 1 à FF). L'adresse de chaque abonné est indépendante de son emplacement physique. Les adresses ne doivent pas obligatoirement être attribuées de manière séquentielle. Deux abonnés ne peuvent avoir la même adresse.

3. Format de la trame d'échange requête/réponse

3.1. La requête

Elle contient un code fonction indiquant à l'esclave adressé quel type d'action est demandé. Les données contiennent des informations complémentaires dont l'esclave a besoin pour exécuter cette fonction.

N° d'esclave	Code Fonction	Information spécifique concernant la demande	CRC
1 octet	1 octet	n octet	2 octets

La trame de requête contient aussi un code de vérification d'erreur CRC permettant à l'esclave d'accepter le message ou de le rejeter.

3.1. La réponse

N° d'esclave	Code Fonction	**Données reçues**	CRC
1 octet	1 octet	n octet	2 octets

Suite à la requête émise par le maitre, l'esclave répond et envoie les données demandées.

Si une erreur apparaît, le code fonction est modifié pour indiquer que la réponse est une réponse d'erreur. Les données contiennent alors un code (code d'exception) permettant de connaître le type d'erreur.

4. Codage des octets de la trame

Deux types de format de trames peuvent être utilisés pour communiquer sur un réseau Modbus. Tous les équipements présents sur le réseau doivent être configurés selon le même type.

4.1. Codage ASCII

Chaque octet composant une trame est codé avec 2 caractères ASCII (American Standard Code for Information Interchange)

4.2. Codage RTU (Remote Terminal Unit)

Chaque octet composant une trame est codé sur 2 caractères hexadécimaux (2 fois 4 bits).

Remarque : Le mode ASCII permet d'avoir des intervalles de plus d'une seconde entre les différents caractères sans que cela ne génère d'erreurs, alors que le mode RTU permet un

débit plus élevé pour une même vitesse de transmission. Il est à noter que 95% des modules communicant sur Modbus utilisent des trames RTU.

5. Exemple d'échange entre un maitre et un esclave

Dans ce qui suit, nous allons illustrer un exemple d'échange entre un maitre et un esclave via le réseau Modbus, on considérera deux cas de figures, une transmission sans erreur et une transmission avec erreur

Trame émise par le maître : 04 03 00 02 0001 25 CA

- Adresse esclave : 04

- Code fonction 03 = lecture registre

- N° du registre de début de lecture : MSB : 00 et LSB : 02

- Nombre de registre de lecture : MSB : 00 et LSB : 01

- CRC : 25 CA

Réponse de l'esclave avec erreur : 04 83 02 01 31

- Adresse esclave : 04

- Code fonction : lecture avec MSB = 1 : 83

- Code erreur (n° registre) : 02

- CRC : 01 31

Réponse de l'esclave sans erreur : 04 03 02 02 58 B8 DE

- Adresse esclave : 04

- Code fonction : lecture registre : 03

- Nombre d'octets données : 02

- Données du registre 0002 : MSB 02 et LSB : 58

- CRC : B8 DE

5. Couche physique

Le protocole Modbus peut être implémenté :

- sur une liaison série asynchrone de type RS-232, RS-422 ou RS-485 ou TTY (boucle de courant), avec des débits et sur des distances variables ;
- via TCP/IP sur Ethernet ; on parle alors de Modbus TCP/IP ;
- via Modbus Plus. Modbus Plus est un réseau à passage de jetons à 1 Mb/s, pouvant transporter les trames Modbus et d'autres services propre à ce réseau.
- Une liaison multipoints de type RS-485 relie maître et esclaves via une paire différentielle qui permet un débit élevé (jusqu'à 10 méga-bits par seconde) sur une distance importante (jusqu'à 1 200 m). Elle ne dispose que de 2 bornes qui alternativement passent les données dans un sens puis dans l'autre (half duplex).

6. Services supportés par MODBUS

MODBUS offre 19 fonctions différentes. Elles se caractérisent par un code fonction sur un octet (en hexadécimal). Tous les équipements ne supportent pas tous les codes fonction.

Code	**Fonction Modbus**
H'01'	Lecture de n bits de sortie consécutifs
H'02'	Lecture de n bits d'entrée consécutifs
H'03'	Lecture de n mots de sortie consécutifs
H'04'	Lecture de n mots consécutifs d'entrée
H'05'	Ecriture de 1 bit de sortie
H'06'	Ecriture de 1 mot de sortie
H'07'	Lecture du statut d'exception
H'08'	Accès aux compteurs de diagnostic
H'09'	Téléchargement, télé déchargement et mode de marche
H'0A'	Demande de CR de fonctionnement
H'0B'	Lecture du compteur d'événements
H'C'	Lecture des événements de connexion
H'D'	Téléchargement, télé déchargement et mode de marche
H'E'	Demande de CR de fonctionnement
H'F'	Ecriture de n bits de sortie
H'10'	Ecriture de n mots de sortie
H'11'	Lecture d'identification
H'12'	Téléchargement, télé déchargement et mode de marche

H'13'	Reset de l'esclave après erreur non recouverte

Tableau C.I.1. Les services supportés par Modbus [B9]

C.II. Réseau Lon

1. Introduction

LonWorks® est un standard de communication ouvert conçu pour le monde de la GTB, ce réseau ouvert a été développé par la société Américaine Echelon, il a été conçu de façon à être utilisé aussi bien avec les solutions d'automatisations centralisées qu'avec les solutions décentralisées. Ce système de communication a été adopté par des normes internationales (ANSI/CEA-709.1-B and ISO/IEC DIS 14908), il permet a des équipements intelligents de communiquer entre eux via un réseau local, LON est l'acronyme de *Local Operating Network*.

LonWorks est considéré comme l'un des premiers standards ouverts à être proposé sur le marché, un standard ouvert permet d'intégrer des composants de différents constructeurs, ainsi, le client ne sera pas dans l'obligation de se restreindre à un seul fabricant lors du choix des composants pour une solution d'automatisation [B8].

1.1. Aperçu historique de LonWorks®

La Société Echelon a été crée en 1986 en Californie par Mike Markkula, le cofondateur de la société Apple, le but était de développer un système ouvert pour les réseaux décentralisés, ce qui a donné naissance a LonWorks ou Lon comme abréviation, au cœurs de cette technologie, on trouve un microcontrôleur révolutionnaire, à savoir le 'Neuron Chip' (ou la puce neurone) baptisé ainsi à cause de l'analogie avec le neurone du cerveau humain.

E 1997, on a enregistré une percée technologique de la société Echelon, en effet, la commercialisation de LonWorks Network Services, une plate forme de développement pour les applications ouvertes et les outils d'intégration, a permis à LonWorks de s'imposer dans le marché des standards de communication ouverts [B8].

2. Eléments du système LonWorks

2.1. Composants de la technologie LonWorks

La technologie LonWorks consiste en vaste nombre d'éléments interconnectés, incluant des composants hardware, des logiciels (software) ainsi que des structures d'organisation, on distingue donc les éléments suivants :

2.1.1. Le Neuron Chip

Le Neuron Chip, développé par la société Echelon, est composé de trois processeurs effectuant des fonctions de communication et de traitement de l'information, le nom de neurone a été choisi à cause des similarités entre la topologie du composant et celle du neurone humain. Le Neuron Chip représente un nœud dans un réseau.



Fig. C.II.1. Le Neuron Chip [B8]

2.1.2. Le protocole LonTalk

Le protocole LonTalk définit la méthode de programmer les Neuron Chips pour différentes applications et la manière par laquelle ces composants communiquent entre eux dans un réseau, le protocole LonTalk est directement embarqué sur les Neuron Chips comme microprogramme permettant ainsi une compatibilité entre tous les nœuds d'un même réseau.

2.1.3. L'émetteur/récepteur

L'émetteur récepteur (transceiver en anglais) joue le rôle d'une interface de communication physique entre les composants LonWorks et le réseau LonWorks, on trouve des émetteurs/récepteurs spécifique pour chaque type de média de communication (la paire torsadée, la fibre optique, les ondes radio…etc.)

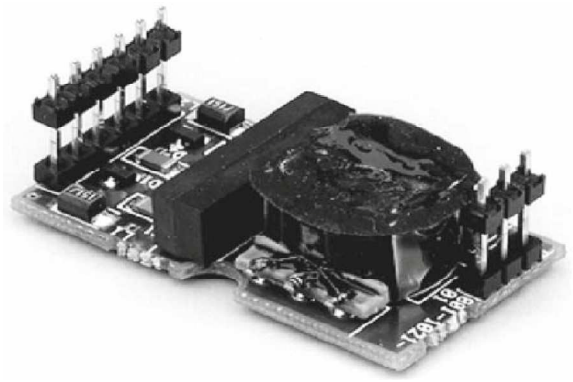


Fig. C.II.2. Un émetteur/récepteur pour connecter un nœud à un réseau [B8]

2.1.4. Les outils de LonWorks

La société Echelon propose divers outils de programmation et d'intégration pour la technologie Lon, on trouve les outils de développement à l'instar de LonBuilder et NodeBuilder permettant de programmer ses propres applications sur le Neuron Chip, on trouve aussi d'autres outils de réseau tel que LonMaker permettant de personnaliser et de paramétrer les composants Lon afin de les intégrer dans un réseau fonctionnel.

3. Fonctionnement des équipements LonWorks

Les différents équipements d'un système Lon sont illustrés dans la figure ci-dessous, on distingue le media (ou support physique de communication : paire torsadée, la fibre optique…etc.) l'émetteur/récepteur (transceiver), les mémoires externes, l'adaptateur de puissance ainsi que les cartes de conditionnements des signaux E/S

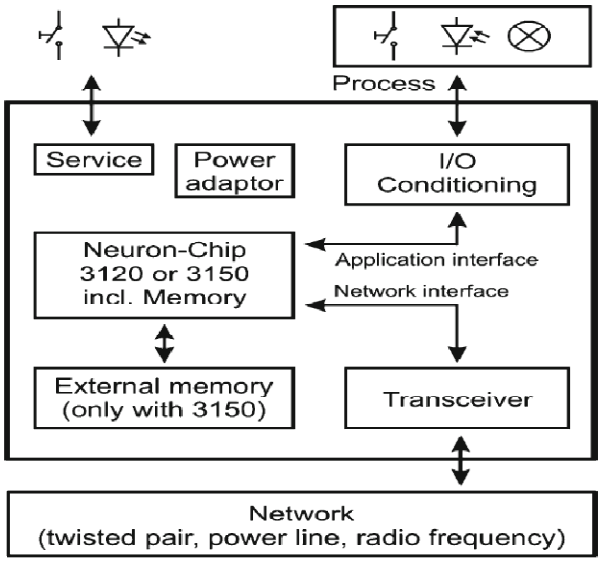


Fig. C.II.3. Equipements d'un système Lon [B8]

3.1. Caractéristiques du Neuron Chip

Le Neuron chip représente l'élément intelligent dans un système Lon, il exécute le protocole LonTalk et sa mémoire associée sauvegarde les programmes décrivant le fonctionnement du composant, il existe deux type de ce composant ; le Neuron Chip 3120 et le 3150, ce dernier est équipée d'une mémoire externe pour sauvegarder des algorithmes plus sophistiqués. La figure ci-dessous illustre les différents composants internes du Neuron Chip

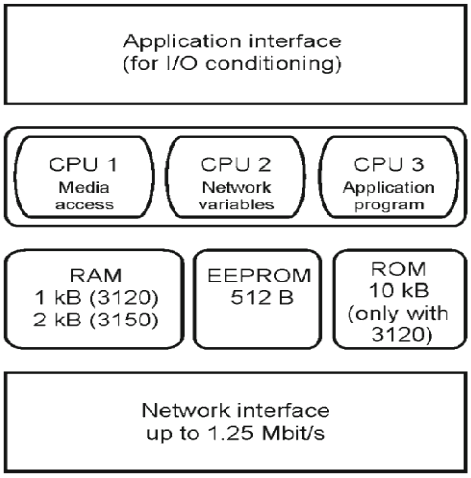


Fig. C.II.4 Composants internes du Neuron Chip [B8]

Le Neuron Chip contient donc trois processeurs effectuant différentes taches, le premier ; CPU 1, est chargé d'accéder au media de transmission et ceci à travers une interface réseau qui n'est rien d'autre que l'émetteur/récepteur, ceci matérialise donc la couche 1 (couche physique) et la couche 2 (couche liaison de données) du modèle OSI.
Le CPU 2 est responsable de transmettre les variables à travers le réseau et matérialise donc les couches de 3 à 6 du modèle OSI.

Le CPU 3 est responsable de l'exécution des programmes d'application, il n'accède donc pas au réseau.
La EEPROM sauvegarde les programmes d'application et les paramètres de configuration.
La ROM, quand à elle, sauvegarde le protocole LonTalk. Pour le Neuron Chip 3150, la sauvegarde du protocole se fait dans une mémoire externe contenant un espace supplémentaire pour sauvegarder des algorithmes sophistiqués.

3.2. Conditionnement des Entrées Sorties

Afin d'être traité par le processeur, les signaux provenant du process doivent subir certains traitements, on parle alors du conditionnement des signaux d'entrées sorties, par exemple, un signal analogique provenant d'un capteur de température de sortie résistive doit être convertit en un signal numérique en passant par un convertisseur analogique numérique pour que le Neuron Chip puisse traiter ce signal.

3.3. Unité d'alimentation et adaptateur de puissance

La puissance nécessaire pour le fonctionnement des composants provient soit d'une source d'énergie externe ou bien directement prélevée du bus de communication lorsque le composant ne consomme pas une grande quantité d'énergie, une autre possibilité consiste à prélever l'énergie directement du réseau électrique alimentant le bâtiment, il est à noter que le réseau électrique (de puissance) peut aussi servir à transmettre les données. L'adaptateur de puissance est requis pour convertir et stabiliser la tension électrique.

3.4. Types et caractéristiques des émetteurs/récepteurs

La fonction d'un émetteur/récepteur (en anglais transceiver contraction de transmitter/receiver) est de connecter le Neuron Chip avec le réseau et ceci à travers une interface ayant un débit pouvant atteindre 1.25 Mbit/s. Ainsi, les différents types d'émetteurs/récepteurs permettent de relier la même puce (Neuron chip) à des médias différents (la paire torsadée, le réseau de puissance…etc.), pour les émetteurs/récepteurs fonctionnant sur un réseau d'alimentation électrique (ou PLT pour Power Line Transceivers), des problèmes peuvent apparaitre à cause des bruits générés par les charges branchées sur le réseau, à fin de limiter l'impact de l'interférence, le débit de transmission est limiter à 5Kbit/s.

Le tableau suivant montre les différents types des émetteurs /récepteurs ainsi que leurs caractéristiques.

Media	**Emetteur/récepteur**	**Débit**	**Topologie**	**Longueur du Réseau**	**Alimentation**
Paire torsadée	FTT-10A	78 Kbit/s	Topologie libre, Bus	500 m, 2700m	Séparée
Paire	LPT-10	78 Kbit/s	Topologie	500 m,	A travers le

torsadée			libre, Bus	2700 m	bus
Paire torsadée	TPT/XF-78	78 Kbit/s	Bus	1400 m	Séparée
Paire torsadée	TPT/XF 1250	1.25 Mbit/s	Bus	130 m	Séparée
Réseau Electrique	PLT-22	5 kbit/s	Topologie libre	*	A travers un adaptateur de puissance

* Dépend de l'atténuation et l'interférence

Tableau. C.II.1 Types des émetteurs/récepteur [B8]

Désignations :

FTT : *Free Topology Transceiver* ou émetteur/récepteur à topologie libre

LPT : *Link Power Transceiver* ou émetteur/récepteur avec liaison d'alimentation (permettant d'alimenter l'émetteur/récepteur directement du bus de communication).

TPT : *Twisted Pair Transceiver* ou émetteur/récepteur sur paire torsadée (le XF désigne le transformateur utilisé pour isoler le composant du réseau)

PLT : *Power Line Transceiver* ou émetteur récepteur branché sur le réseau électrique

4. Transfert d'information entre les composants de LonWorks

Dans cette section, nous allons montrer comment créer un réseau à partir des composants Lon, la première étape consiste à établir les connections **physiques** entre les éléments du réseau en respectant la topologie relative à chacun des émetteurs/récepteurs, en deuxième étape, les connections **logiques** entre les éléments du réseau doivent être établies, ceci permettra aux composants de communiquer en utilisant les variables des programmes d'application.

4.1. Topologies des réseaux physiques

Dans ce qui suit, nous allons montrer les principales topologies de réseaux utilisées dans la technologie LonWorks pour connecter les nœuds du réseau, et comme on l'a déjà mentionné, un nœud est matérialisé par le processeur Neuron Chip.

4.1.1. Topologie en Bus

Cette topologie permet d'avoir des réseaux de longues distances, ainsi, en utilisant des émetteurs/récepteurs de types FTT ou LPT, on peut atteindre des distances de 2700 m, le câble liant les nœuds au bus ne doit pas dépasser 3 m de longueur. Afin d'éviter le phénomène de réflexion aux extrémités du câble, on utilise des terminaisons T ayant comme résistance une valeur de 107 Ω

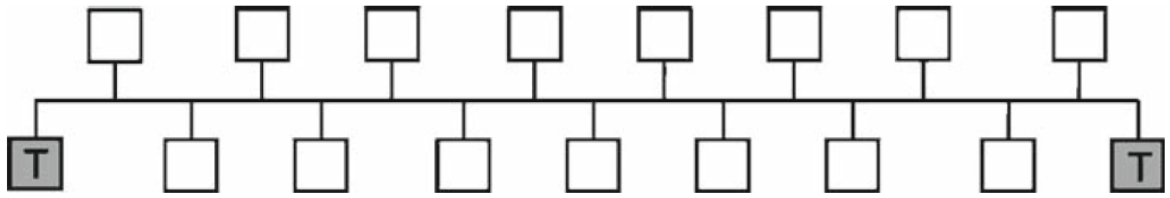


Fig. C.II.5. Nœuds reliés au bus avec des terminaisons [B8]

4.1.2. Topologie en étoile et topologie en anneau

Les topologies en étoile et en anneau sont implémentées en utilisant des émetteurs/récepteur à topologie libre, pour ces deux topologies, la distance maximale est de 500 m, de plus, la distance entre deux nœuds ne peut excéder 320 m. La résistance des terminaison (T) est de 52.3 Ω, il est à noter que la topologie en anneau est la plus appropriée pour connecter des composant Lon dans les chambres et ceci grâce à sa flexibilité et la facilité de mise en œuvre et de modification.

4.1.3. Les sous réseaux (subnets)

Le sous réseau est une partie du réseau global et contient un certain nombre d'équipements. Il peut contenir un maximum de 128 nœuds adressables, ce chiffre de 128 est atteint lorsqu'on utilise des émetteurs/récepteurs types LPT, le nombre maximum est de 64 pour des émetteurs/récepteurs types FTT. Les composants Lon travaillant ensembles doivent être mis dans le même sous réseau pour minimiser le temps d'exécution des commandes.

- ***Les répéteurs***

Les répéteurs sont utilisés lorsqu'on veut connecter plus de 64 composants (le nombre maximum) à un réseau utilisant des émetteurs récepteurs types FTT, le nombre de répéteurs installés dans une ligne ne doit pas dépasser trois, sinon des retards de transmission peuvent apparaitre.

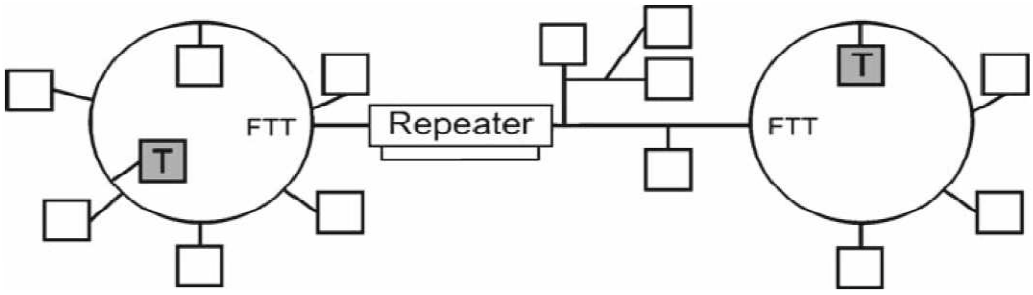


Fig. C.II.6. Sous réseau avec répétiteur (repeater) [B8]

- ***Les routeurs et les canaux***

Les routeurs sont utilisés pour connecter des segments (sous réseaux ou encore canaux dans la terminologie de Lon) utilisant différents médias physiques, une différence entre les routeurs et les répéteurs est que les routeurs peuvent filtrer les trames en utilisant une table de routage pour ne faire passer que les données dont l'adresse est dans le prochain segment (évitant ainsi de surcharger l'arête dorsale (backbone) par des données qui sont destinées à un destinataire sur le même segment).

La figure III.7 illustre l'exemple d'un routeur utilisé pour connecter deux canaux (segments) utilisant des médias différents (celui à droite utilise le réseau électrique pour la transmission de données tandis que celui à gauche utilise des émetteurs/récepteurs type FTT donc de la paire torsadée comme média)

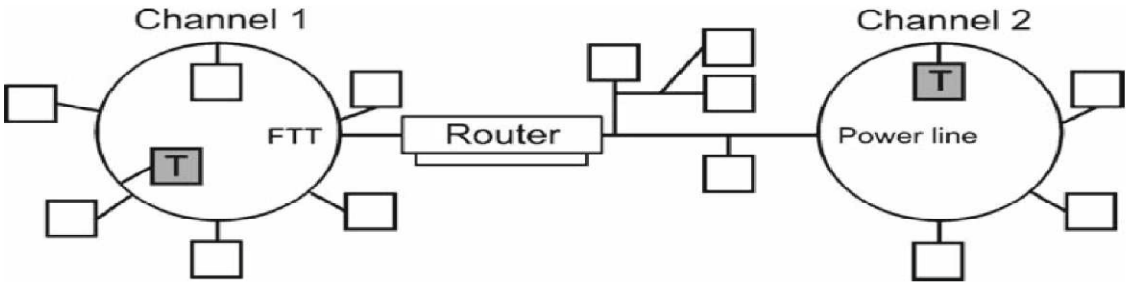


Fig. C.II.7 Routeur pour connecter deux canaux [B8]

4.1.4. Les domaines

Un domaine consiste en un nombre de sous réseaux (subnets) connectés via un routeur, en se référant au protocole LonTalk, un domaine peut comporter jusqu'à 255 sous réseaux, chacun des sous réseaux peut comporter un maximum de 128 nœuds (127 Neurons plus le routeur), ce qui donne un maximum de 32,385 composants adressables dans un domaine.

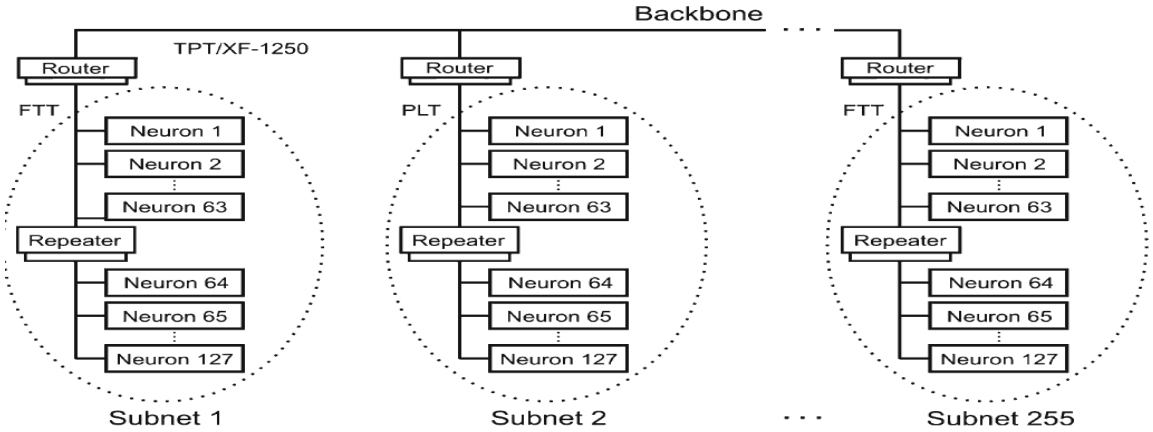


Fig. C.II.8 Sous réseaux reliés pour former un domaine [B8]

4.2. Codage du signal

Pour la transmission des données, on utilise le **code Manchester différentiel** dans lequel l'envoi de données se fait de façon synchronisée avec une horloge ayant une fréquence donnée (la valeur de cette fréquence est déterminée par le débit de transmission). La figure III.9 illustre le principe de ce code; un 0 logique est reconnu en détectant une transition du signal entre deux top d'horloge, sinon c'est le 1 logique qui a été envoyé.

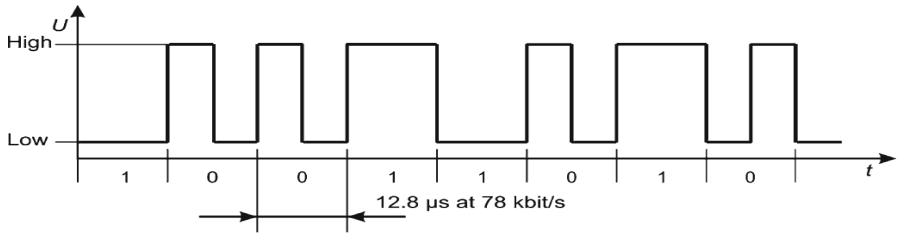


Fig. C.II.9 Code Manchester différentiel [B8]

4.3. Architecture du réseau logique et variables du réseau

Le principe d'échange d'informations utilisé par le protocole LonTalk™ repose sur la notion de variable. Une variable représente généralement une information unitaire (une valeur, un état, etc.).

Une des clés de l'interopérabilité des systèmes dans un réseau Lon est la standardisation des variables utilisés pour représenter des grandeurs physiques, ces variables standards sont connus sous le nom de SNVT (pour Standard Network Variable Type) et sont publiés par LonMark International, l'organisation chargée de promouvoir l'utilisation de la technologie LonWorks et de certifier les équipements répondant aux standards LonWorks.

Ainsi par exemple un thermostat utilisant la variable SNVT température produirait un nombre entre zéro et 65535 pour exprimer une température entre -274 et 327 degrés Celsius

Les capteurs "publient" les informations sur le réseau à travers des variables de sortie. On dit qu'ils "produisent" la variable. Les actionneurs "s'abonnent" aux informations dont ils ont besoin à travers des variables d'entrée. On dit qu'ils "consomment" la variable.

Les relations producteur /consommateur ne sont pas prédéterminées, elles sont définies à l'aide d'un outil de configuration réseau. De même, ces relations ne sont pas immuables, elles peuvent être ajoutées, étendues ou supprimées dès que cela s'avère nécessaire (modification ou extension du site) et ce, sans interrompre le fonctionnement du réseau et donc de l'installation.
Les variables sont typées c'est à dire qu'en fonction du type de l'information, un format standard est appliqué. Ce format international est à la base de l'interopérabilité. Il permet, en effet, d'assurer que des produits d'origines diverses puissent "comprendre" les informations qu'ils s'échangent. Ainsi un capteur de température intelligent d'origine européenne pourra

communiquer avec un automate/régulateur fabriqué aux Etats-Unis car tous deux expriment une température de la même manière. Les types de variables standardisés sont appelés des SNVTs (Standard Network Variable Type).

Le tableau suivant présente quelques exemples des types de variables (SNVTs) communément utilisées en GTB

Type	**Information**	**Valeur**	**Application**
SNVT_temp_p	Température	-273 à 327° Celsius (0,01 °C)	Température mesurée/de consigne
SNVT_hvac_modeSNVT_hvac_status	Mode de fonctionnement	0 = HVAC_AUTO choix du mode automatique 1 = HVAC_HEAT Chauffage seulement 2 = VAC_MRNG_WRMUP démarrage matinal 3 = HVAC_COOL rafraîchissement seulement 4 = HVAC_NIGHT_PURGE purge de nuit 5 = HVAC_PRE_COOL pré-rafraîchissement 6 = HVAC_OFF Hors fonctionnement 7 = HVAC_TEST Auto-test 8 = HVAC_EMERG_HEAT chauffage d'urgence 9 = HVAC_FAN_ONLY ventilation seulement	Commande/contrôle du mode de fonctionnement du régulateur de climatisation
SNVT_lev_disc	Niveau	Valeurs discrètes suivantes : 0 = rien / arrêt 1 = bas 2 = moyen 3 = haut 4 = plein	Niveau de puissance, de vitesse, etc.
SNVT_lev_percent	Niveau, pourcentage	0 à 100 % (0.5%)	Humidité relative, % d'ouverture de vanne
SNVT_occupancy	Présence	Valeurs discrètes suivantes : 0 = zone occupée 1 = zone inoccupée 2 = zone temporairement occupée 3 = zone temporairement inoccupée	Détecteur de présence, Programme horaire

SNVT_switch	Commutation	Information composite : Etat : 0 = Arrêt 1= Marche Valeur : 0-100% (0,5%)	Vitesse de ventilation E/S logiques
SNVT_flow	Débit	0 - 65534 litres/sec (1 l/s)	Débit d'air (VAV), de liquide

Tableau C.II.3. Types de variables SNVT communément utilisés en GTB [B8]

III. Réseau BACnet

1. Introduction

BACnet (Building Automation and Control Network) est un Protocol standard de communication dédié à l'automatisation des bâtiments et développé par la société américaine de chauffage, réfrigération, et conditionnement d'air (ASHRAE).Par la suite c'est le groupement d'intérêt BACnet International qui va promouvoir l'utilisation de ce Protocol.

BACnet est apparu comme une solution de communication et d'échange de données entre les divers constituants d'un système d'automatisation et ceci indépendamment des constructeurs ce qui assure une interopérabilité entre les équipements de divers fabriquant.

Une bonne stratégie pour minimiser les dépenses consiste à utiliser le réseau LAN déjà présent dans le bâtiment pour transporter les *messages BACnet*, on peut par exemple utiliser le réseau Ethernet, ARCnet ou bien MS/TP.

Le protocole BACnet se base sur ce qu'on appelle **objets**. A chaque élément intervenant dans l'automatisation du bâtiment (entrée/sortie, variables de contrôle, paramètres du régulateur…) est affecté d'un objet qui possède certaines propriétés (ex : nom, état courant…etc.). Ces objets permettent de recevoir des informations pour chaque dispositif sans pour autant connaitre son fonctionnement interne ou sa configuration. Les *objets standardisés* sont donc la clé de l'interopérabilité des équipements de divers fabriquant.

2. Architecture de communication BACnet

BACnet a été développé suivant le model de référence OSI. La **figure C.III.1** montre la correspondance entre les diverses couches du model OSI et celles de BACnet :

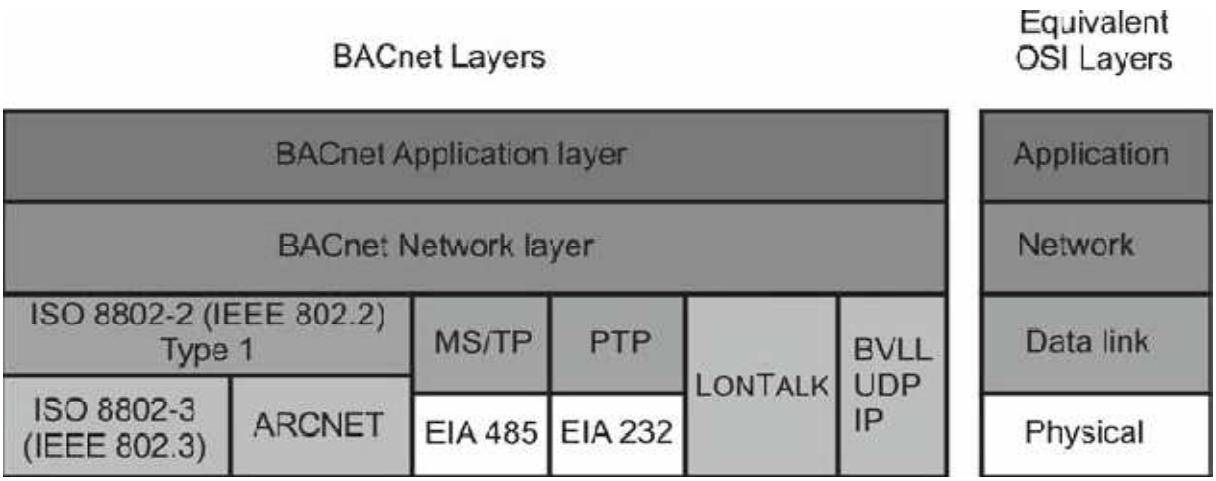


Fig. C.III.1. Couches et protocoles du réseau BACnet [B8]

Les fonctions des couches présentation, session et transport sont intégrées dans la couche application de BACnet.

Les réseaux BACnet sont des réseaux LAN où les dispositifs sont **statiques** (c.à.d. qu'ils exécutent toujours les mêmes opérations).

La couche physique fournit les moyens de connections des dispositifs BACnet ainsi le transport des signaux à travers les médias. La couche Liaison de donnée quand à elle, organise les données sous forme de trames, régule le transfert et le flux de données sur le média et détecte les erreurs de transmission.

La plupart des fonctions de la couche Transport sont similaires aux fonctions de la couche Liaison de données (segmentation des données, control du flux…) mais différent par leurs portées. En effet la couche liaison de donnée se focalise sur les connections *point à point* entre deux équipements dans un même réseau alors que la couche transport traite des connections entre deux extrémités limites à travers des réseaux multiples. Les fonctions de la couche transport sont effectuées par la couche application de BACnet.

Les différentes étapes d'encapsulation des données BACnet pour former les messages à envoyer sur le média sont illustrées par la **figure C.III.2** :

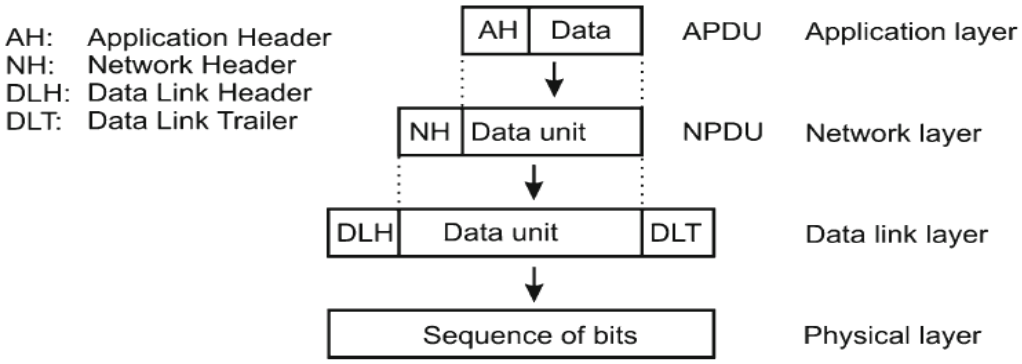


Fig. C.III.2. Encapsulation des données [B8]

Un en-tête d'application (AH :Application Header) est ajoutée à la donnée originelle à envoyer (demande de manipulation des objets) formant ainsi un **APDU** (Application Protocol Data Unit). La couche réseau à son tour rajoute une en-tête (NH : Network Header) qui contient les adresses réseau globales formant ainsi un **NPDU** (Network Protocol Data Unit) qui sera ensuite passée à la couche liaison de données qui rajoute une en-tête DLH (Data Link Header) contenant les adresses réseau Locales et un en-queue.

3. Couches physique et Liaison de données de BACnet

Comme on l'a dit précédemment, BACnet supporte plusieurs types de réseaux LAN mais c'est au concepteur de choisir le plus approprié en fonction du cout et des performances souhaités. Les réseaux LAN les plus utilisés pour BACnet sont donnés par la **figure C.III.3**

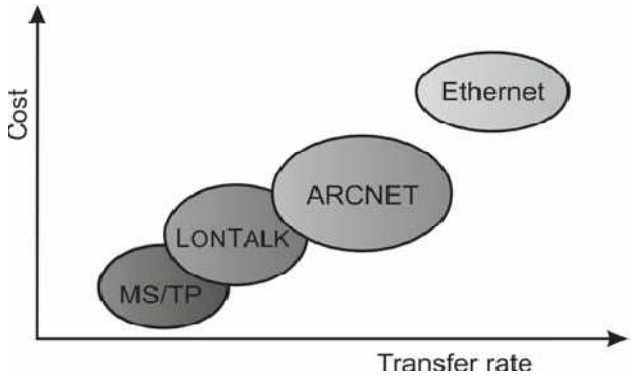


Fig. C.III.3 Comparaison des réseaux LAN utilisés pour les deux premières couches de BACnet [B8]

Dans ce qui suit nous allons discuter des réseaux MS/TP et ARCNET vu qu'Ethernet a déjà été abordé précédemment et LONTALK sera traité dans le chapitre dédié à LONWORKS.

3.1 Master-Slave/Token-Passing (MS/TP), EIA-485 et EIA-232

Ce protocole est simple pas cher à mettre en œuvre et il est bien adapté aux petites unités de contrôle où on n'a pas besoin d'une vitesse de transmission élevée. Un réseau MS/TP utilise pour la transmission la paire torsadée blindée (STP) avec une distance maximale de 1200m pour le segment. La couche physique est basée sur le standard EIA-485 (RS-485) qui utilise une communication série des données.

3.1.1 La couche physique

Les données au niveau de cette couche sont transférées en octets avec un intervalle variable entre deux octets successifs (communication asynchrone). La **figure C.III.4** illustre cette transmission :

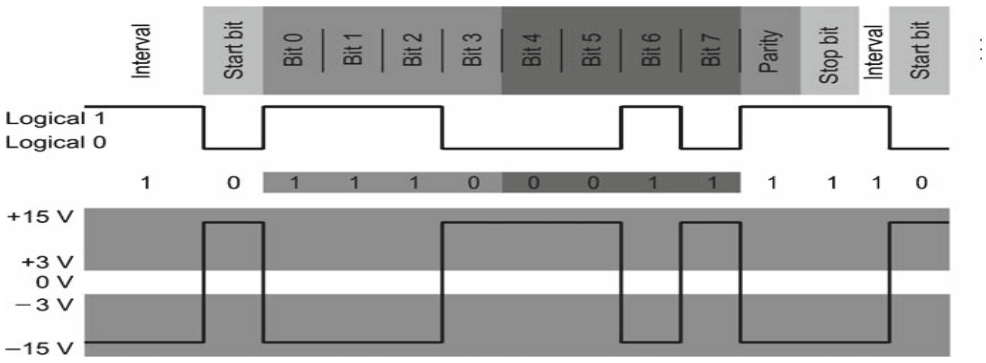


Fig. C.III.4. Méthode de transmission des données [B8]

Le débit moyen (standard) est d'environ 9,600 Bd(mais qui peut aller jusqu'à 76,800 Bd). Ce faible débit et la simplicité de ce protocole permet de l'implémenter sur microcontrôleurs à faible cout.

Il y a deux grandes différences entre EIA-485 et l'interface série du PC EIA-232(RS-232) :

- EIA-232 est à voltage référencé par rapport à la terre c.à.d. que les bits sont convertis en volts (si le niveau est compris entre -3V et -15V par rapport à la terre alors cela représente un 1 logique et si le niveau est compris entre +3V et +15V par rapport à la terre, cela équivaut à un 0 logique).En d'autre termes il ne faut que 3 câbles pour relier deux équipements comme le montre la **figure. C.III.5**

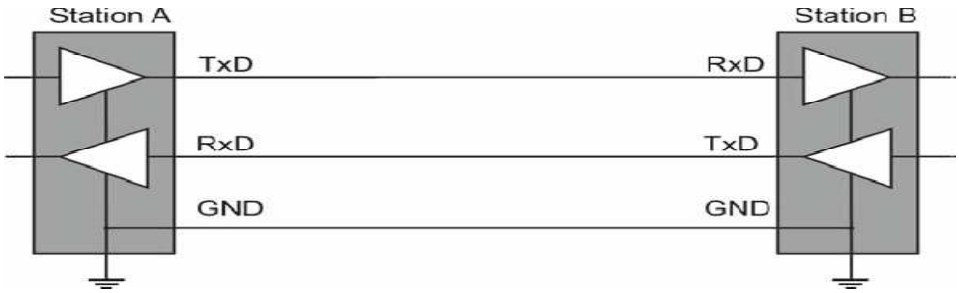


Fig. C.III.5. RS-232 :TxD(transmission),RxD(Réception) [B8]

- EIA-485 est une interface à voltage différentiel comme le montre la **figure. C.III.6**

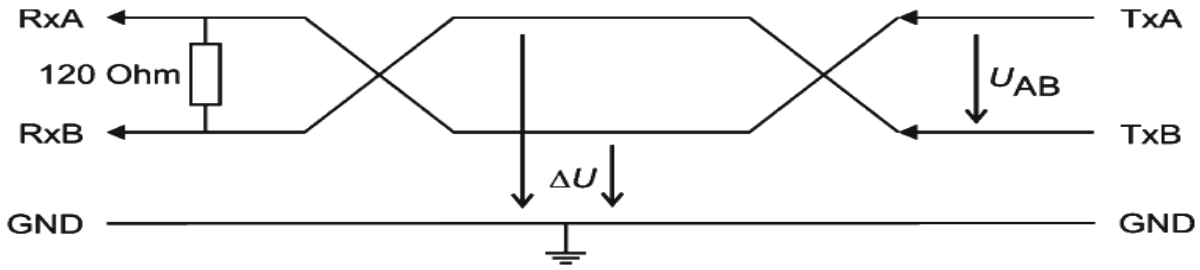


Fig. C.III.6. Transmission avec RS-485 [B8]

Le transmetteur applique une tension U_{AB}=±3 V à 6V entre les deux câbles A et B (Une tension positive de U_{AB} représente un 1 logique alors qu'une tension négative représente un 0 logique). Si des interférences apparaissent le long du câble, cela conduit à une augmentation du volage de ΔU par rapport à la terre et pour chaque câble ce qui conduit à avoir U_{AB} constant malgré la présence de ces interférences.
Cette propriété rend EIA-485 robuste aux interférences et ainsi il peut être utilisé sur de longues distances à l'inverse d'EIA-232 qui lui est utilisé pour de faibles distances (moins de 15 m).

Une autre différence entre EIA-485 et EIA-232 est que ce dernier travaille en full-duplex (transmission simultané des données dans les deux sens) avec une transmission point à point alors que EIA-485 travaille en half-duplex (alternance de la direction de transmission) avec une transmission multipoint pouvant aller jusqu'à 32 nœuds comme le montre la **figure C.III.7**

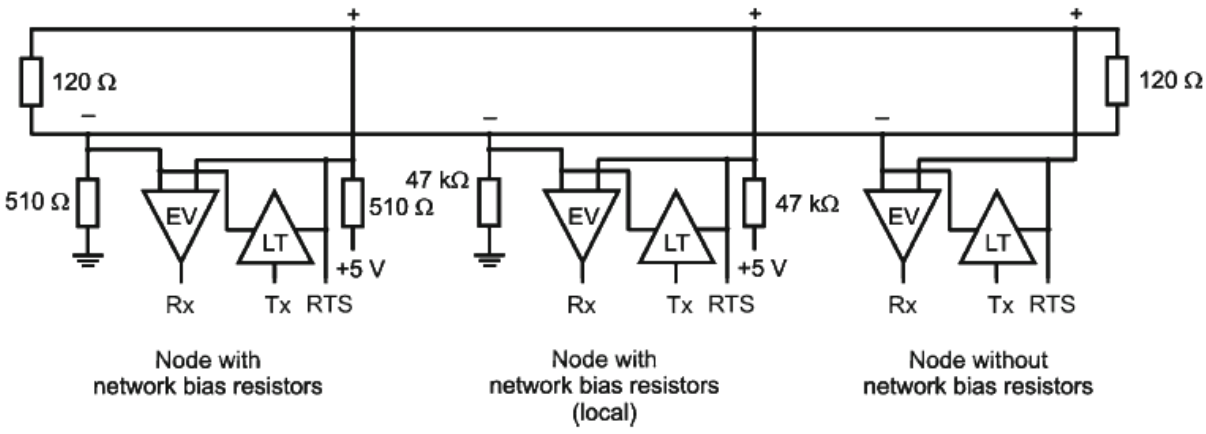


Fig. C.III.7. Réseau utilisant EIA-485 avec 3 types de nœuds [B8]

On voit dans cette figure que chaque segment doit être terminé avec une résistance approprié pour éviter la réflexion du signal.

Chaque segment MS/TP EIA-485 peut être équipé de résistances de polarisation (globales ou locales) qui simulent une tension fixe en absence de signal de transmission. Sans ces résistances, la tension être les deux lignes devient nulle et sensible aux perturbations (qui induisent une différence de tension positive ou négative) ce qui peut fausser la lecture des signaux.

Les signaux reçus doivent être amplifié dés la réception de même que ceux envoyés sur les deux lignes.

La trame à envoyer dans un réseau MS/TP a la structure suivante :

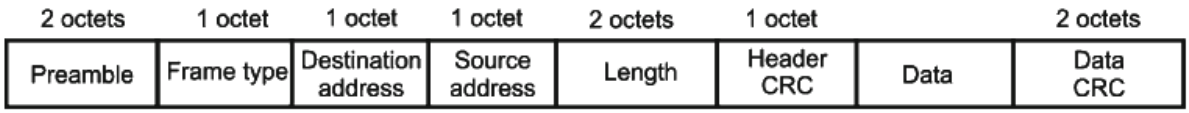


Fig. C.III.8. Format d'une trame MS/TP [B8]

- **Preamble (préambule) :** délimite le début du signal.
- **Frame type :** indique si la trame contient des données ou des informations de contrôle.
- **Source adress et Destination adress :** adresse de l'émetteur et du récepteur respectivement.
- **Length :** cette portion indique la taille de la trame.
- **Header CRC :** en-tête pour la détection d'erreurs.
- **Data CRC :** portion utilisée par le récepteur pour détecter d'éventuelles erreurs.

3.1.2 La couche Liaison de données

Cette couche contrôle l'accès au média grâce au *passage de jeton*. Le jeton (autorisation d'émettre) est passé d'une station à une autre (station à coté).Un réseau MS/TP a des stations maitres et esclaves (Master-Slave). Seule la station maitre peut recevoir le jeton et initier la communication (échange de données) tandis que les stations esclaves ne reçoivent pas de jetons mais attendent simplement une requête.

Les dispositifs de terrain (détecteurs et actionneurs) représentent les esclaves qui sont reliés à un maitre qui peut être par exemple une station d'automatisation. S'il y a plusieurs maitres, celui qui possède le jeton doit le passer à la prochaine station maitre après que soit écoulé un certain intervalle de temps. En d'autre termes, on peut savoir avec exactitude quand est-ce qu'une station va recevoir le jeton c'est pour cela que cette méthode est nommée : *méthode déterministe de contrôle d'accès au média.*

3.2 ARCNET

ARCNET (Attached Resource Computer Network) est un réseau local LAN initialement développé aux USA mais qui a été remplacé par fast Ethernet. À la différence de l'Ethernet, cependant, ARCNET est déterministe ce qui le rend très apprécié dans la communication industrielle. Comme le protocole MS/TP, ARCNET est basé sur le passage de jetons. ARCNET possède les avantages suivants :

- Il fonctionne en temps-réel ce qui permet de prédire avec exactitude le temps de réponse.
- Les couches OSI 1 et 2 sont implémentées à faible cout.

- Il peut s'utiliser avec différentes topologies.
- Il peut s'utiliser avec différents types de médias.
- Le débit varie de 30 bps à 10 Mbps.

Cependant ARCNET n'est pas assez répandu en dehors des frontières Américaines donc peu de dispositifs BACnet sont compatibles avec ARCNET.

4. Couche Réseau de BACnet

4.1 Le but de la couche réseau

Le but de la couche réseau est de transmettre les messages entre des réseaux pouvant avoir différentes couches de Liaison de données (ex : Ethernet et Token ring) et ceci grâce à un routeur comme le montre la **figure C.III.9.**

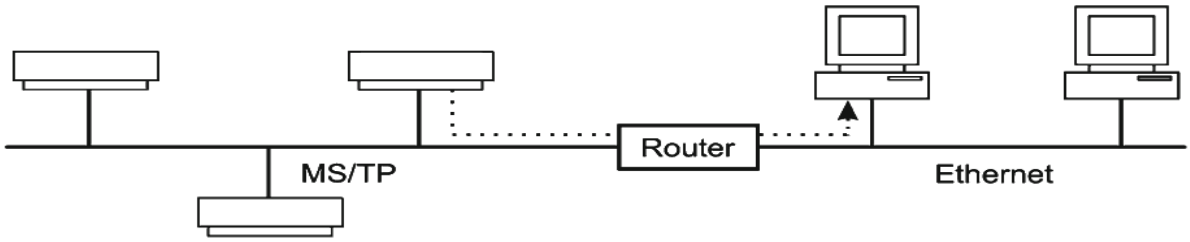


Fig. C.III.9. Deux réseaux connectés via un routeur [B8]

La couche liaison de donnée, comme on l'a vu précédemment ne s'occupe que de l'adressage local d'un réseau mais si on veut interconnecter deux réseaux (2réseaux MS/TP par exemple dont chacun jusqu'à 254 adresse) il faudra les différencier et ceci se fait grâce à une adresse global. Dans les réseaux BACnet, celle-ci est sur 2 octets (max 65 535 sous réseaux). Par conséquent, une station est complètement défini par l'adresse du sous réseau ou elle se trouve et son adresse dans ce sous réseau.

La couche réseau rajoute des en-têtes à la donnée provenant de la couche application (APDU : Application layer Protocol Data Unit) pour donner lui à une nouvelle trame de la couche réseau NPDU(Network layer Protocol Data Unit).

La structure d'un NPDU est donnée par la figure suivante :

Version	1 octet	
Control	1 octet	
DNET	2 octets	DNET: Destination network address
DLEN	1 octet	DLEN: Length of the destination MAC layer address
DADR	Variable	DADR: Destination MAC address
SNET	2 octets	SNET: Source network address
SLEN	1 octet	SLEN: Length of the source MAC address
SADR	Variable	SADR: Source MAC address
Hop Counter	1 octet	
Message type	1 octet	
Vendor ID	2 octets	
DA	*N* octets	DA: Application layer protocol data unit

Fig. C.III.10. Structure d'un NPDU [B8]

- **Version :** version du protocole BACnet
- **Control :** indique s'il s'agit d'un message de routage ou bien d'un message BACnet provenant de la couche application.
- **Hop Counter :** c'est une valeur qui doit être décrémentée et qui représente la durée de vie du message BACnet ce qui permet à ceux-ci de ne pas enter dans une boucle infinie.

4.2 Utilisation des protocoles Internet dans BACnet

BACnet utilise comme protocole de la couche réseau, le protocole IP (Internet Protocol) qui est de plus en plus utilisé dans l'industrie de l'automatisation des bâtiments.

Ce protocole est responsable du routage des paquets à travers des réseaux complexes. Pour cela, ce protocole a besoin des adresses source et destination pour déterminer le meilleur chemin à travers le réseau global.

4.2.1 Les adresses IP

Le protocole IP utilise des adresses structurées (à l'inverse du protocole Ethernet) qui seront utilisées par les routeurs pour déterminer le meilleur chemin à suivre par les paquets à travers les différents réseaux aussi complexes soient ils.

Les adresses IP sont constituées de 32 bits organisés en 4 octets comme le montre la figure A.II.11. Le format décimal est utilisé pour chaque octet pour faciliter la lecture.

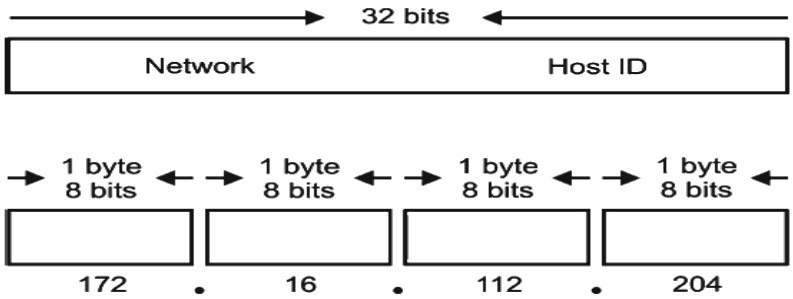


Fig. C.III.11 Format d'une adresse IP [B8]

Cette adresse est divisée en deux parties : l'adresse du réseau et celle de la station hôte. L'adresse de l'hôte correspond à celle dans le réseau local.

Le nombre de bits affectés à chaque type d'adresse varie donnant lieu à 5 différentes classes (selon le nombre de bits affecté à chaque type d'adresse) comme le montre le tableau suivant :

Class	Leading bits	Value of the first byte	Bits for the network number	Bits for the host number	Maximum number of hosts
A	0	1–126	8	24	~16 million
B	10	128–191	16	16	~65,000
C	110	192–223	24	8	~250
D	1110	224–239	Multicast		
E	1111	Reserved for experimental purposes			

Tableau C.III.1. Les différentes classes des adresses IP [B8]

- **Classe A** : cette classe est utilisée pour des réseaux ayant un très grand nombre d'hôtes. Si dans la partie réservée à l'hôte, tous les bits sont à 1 alors l'adresse est dite broadcast c'est-à-dire destinée à l'ensemble des ordinateurs du réseau.

  Pour identifier la partie de l'hôte et celle du réseau un masque réseau de 32(4 octets) bits est introduit. Ce masque contient des 1 dans le champ correspondant à la partie « adresse du réseau » et des 0 dans la partie correspondant à la partie « adresse de l'hôte »

- **Classe B:** Ces adresses sont réservées aux réseaux moyens à large.
- **Classe C :** Cette catégorie d'adresse concerne de plus petits réseau et elle est très utilisée par les réseaux LAN.
- **Autres classes :** les autres adresses sont réservées au **multicasting** pour la classe D (envoie des données à un groupe d'ordinateurs) ou pour des raisons expérimentales (classe E).
  Les adresses qui suivent sont à usage privé :

  • 10.0.0.0–10.255.255.255

• 172.16.0.0–172.31.255.255

• 192.168.0.0–192.168.255.255

Elles peuvent être utilisées arbitrairement dans un réseau local mais ne doivent en aucun cas terminer sur Internet ou passé passer par un routeur.

4.3. Processus de routage

Les routeurs utilisent des protocoles de routage pour partager les informations concernant les chemins à suivre avec d'autres routeurs. Ces informations sont utilisées pour construire une table qui mémorise les chemins pour une destination particulière dans le réseau.

L'administrateur du réseau peut entrer des informations dans la table de routage manuellement (routage statique) ce qui a l'avantage de réduire la charge sur le réseau due à l'échange d'informations de routage entre les routeurs. L'inconvénient est que si un chemin est sujet à un défaut, il ne sera pas remplacé automatiquement par un autre comme dans le cas du routage dynamique.

La **figure C.III.12** montre un réseau avec trois routeurs. Un réseau local mono-ordinateur est connecté à chacun des routeurs.

Les réseaux locaux ont comme adresse 10.0.0.0, 11.0.0.0, et 12.0.0.0. Les chemins entre les routeurs sont aussi considérés comme des réseaux et par suite ont une adresse réseau (13.0.0.0, 14.0.0.0, et 15.0.0.0). Chaque port du routeur a sa propre adresse IP et chacun d'eux est marqué soit par E0 (Ethernet) ou bien par S0 ou S1 (connections série).

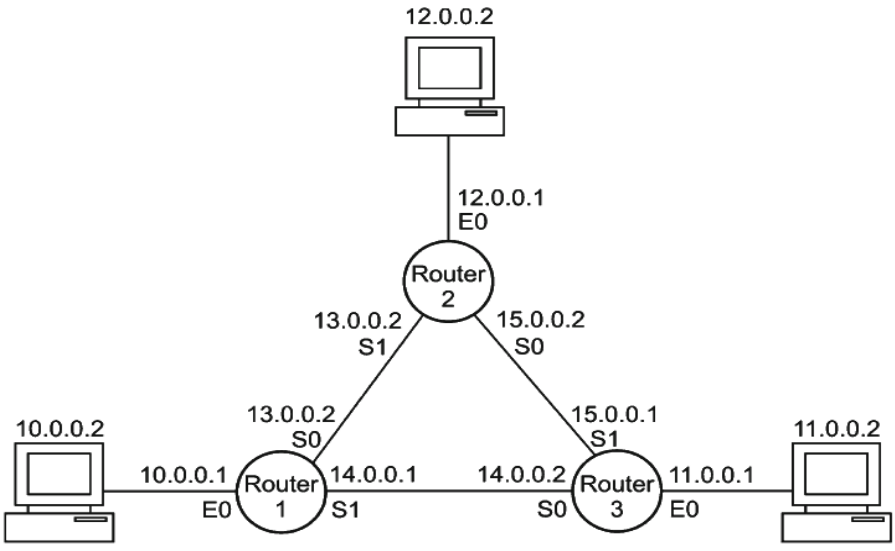


Fig. C.III.12. Réseau comportant trois routeurs [B8]

Les réseaux locaux ont comme adresse 10.0.0.0, 11.0.0.0, et 12.0.0.0. Les chemins entre les routeurs sont aussi considérés comme des réseaux et par suite ont une adresse réseau

(13.0.0.0, 14.0.0.0, et 15.0.0.0). Chaque port du routeur a sa propre adresse IP et chacun d'eux est marqué soit par E0 (Ethernet) ou bien par S0 ou S1 (connections série).

La table de routage est construite comme le montre la ci-dessous :

	Entries for the routing ports		
Network	Router 1	Router 2	Router 3
10.0.0.0	E0	S1	S0
11.0.0.0	S1	S0	E0
12.0.0.0	S0	E0	S1
13.0.0.0	S0	S1	S1
14.0.0.0	S1	S1	S0
15.0.0.0	S1	S0	S1

Fig. C.III.13. Table de routage pour le réseau de la figure A.III.12 [B8]

5. Couche Application de BACnet

Cette couche traite du transfert d'informations de commande, supervision, alertes…etc. Cette information est transférée sous forme d'une séquence de commande codée dans le champ de donnée de l'APDU (Application Protocol Data Unit) comme le montre la **figure A.III.14.**

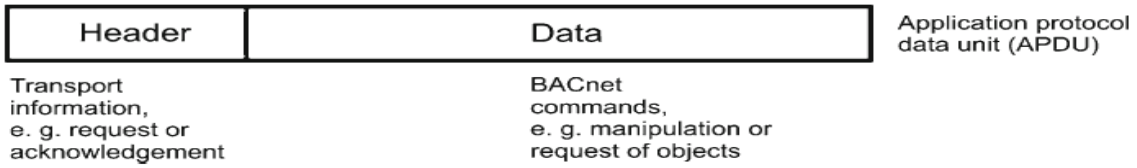


Fig. C.III.14. Structure d'un APDU [B8]

Dans ce qui suit nous allons discuter des *types objet* ainsi que les procédures et services offertes par la couche application.

5.1 Les objets

Les dispositifs BACnet stockent diverses informations telle que la température lu sur un capteur, l'état ouvert ou fermé d'un interrupteur…etc., mais pour un designer de systèmes d'automatisation de bâtiments, il n'est pas important de connaitre comment et ou l'information est stockée dans le dispositif. Ce designer n'a besoin de connaitre que la méthode standardisée d'accès à cette information et c'est pour cela une approche orientée objet a été crée.

Un objet est une structure de donnée dans laquelle l'information est stockée sous forme de propriété de l'objet.

La **figure C.III.15** donne l'exemple d'un objet pour le capteur de température.

Object name	Room temperature
Object type	Analog input
Current value	25.3
Status	Normal
Upper limit	35.0
Lower limit	0.0

Fig. C.III.15. Représentation d'un objet BACnet en utilisant une table [B8]

Le standard BACnet contient divers *types d'objet* qui couvrent l'ensemble de la demande de l'automatisation des bâtiments. On peut distinguer entre les propriétés qui sont nécessaires et lisible (R) , lisibles et ou on peut écrire dessus (W), et propriétés optionnelles (O). On peut aussi combiner les propriétés optionnelles O avec R et W.

Les développeurs peuvent aussi, eux même définir d'autres types d'objets non standardisés ainsi que d'autres propriétés.

5.1.1 Type de données

Les informations stockées dans les propriétés peuvent être de n'importe quel type défini par le tableau de la figure ci-dessous

Data type	Content
NULL	Octet with value 0
Boolean	Boolean (logical) value, TRUE or FALSE
Unsigned integer	Positive whole numbers, for example, with 8, 16, or 32 bit long
Signed integer	Signed integer in two's complement notation, for example, 8, 16, or 32 bit long
Real	IEEE-754 simple precision floating point, for example $1.234 \cdot 10^7$
Double	IEEE-754 double precision floating point
Octet string	String of octets
Character string	String of characters
Bit string	Sequence of bits
Enumerated	List of binary numbers
Date	Date showing the day, month, year and the day of the week
Time	24-h time format

Fig. C.III.16. Types de données pour les objets BACnet [B8]

5.1.2 Conventions concernant les noms d'objets et leurs adressages

Chaque objet Dans BACnet est référencé par un unique identifiant sur 32 bits dont 10 bits pour le type d'objet, comme le montre la figure ci-dessous

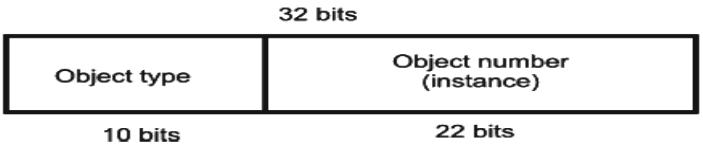


Fig. C.III.17. Identifiant de l'objet BACnet [B8]

Cet identifiant fait partie des propriétés de l'objet. Quand un message est envoyé, l'identifiant de l'objet doit être inclut dans le champ de donnée de l'APDU. Chaque dispositif BACnet possède un objet pour le dispositif lui-même qui a aussi un identifiant d'objet.

Malgré le fait que l'identifiant de l'objet est très utile pour accéder aux objets, cependant, les utilisateurs ainsi que les programmeurs préfèrent travailler avec des *noms.* C'est pour cette raison que chaque objet BACnet est doté d'un nom d'objet qui doit être unique dans tout le réseau.

5.1.3 Types d'objets standards

Pour avoir une idée des types d'objets utilisés dans BACnet, nous allons expliciter le type «ANALOG_INPUT Object » ainsi que ces propriétés. Les autres types d'objets sont disponibles dans la fiche technique de BACnet.

Ce type d'objet est standardisé et représente la propriété d'une entrée analogique telle que celle issue d'un capteur de température. Si un dispositif BACnet possède plus d'une entrée analogique alors ces dernières partagent le même type d'objet mais différent de part l'identifiant de l'objet. Les propriétés de l'objet peuvent être requises par un autre dispositif comme le montre la figure suivante :

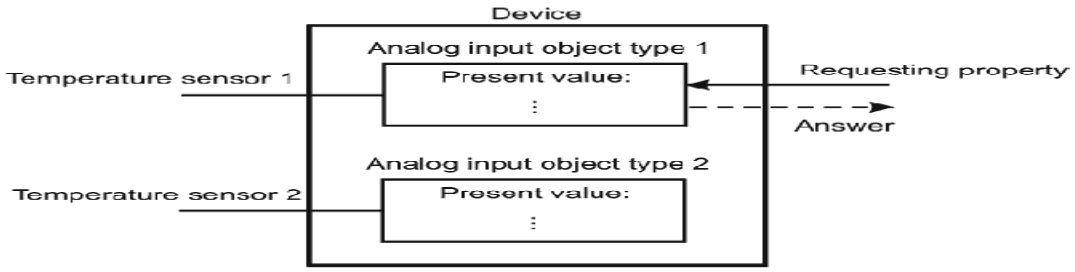


Fig. C.III.18. Représentation des objets dans un dispositif BACnet [B8]

Le tableau de la figure suivant donne un exemple des propriétés d'un objet :

- **Object_Identifier :**code numérique unique sur 32 bits utilisé pour identifier l'objet.
- **Object_Name :** nom de l'objet.
- **Object_Type :** Type de l'objet.
- **Present_Value :** valeur courante mesurée.
- **Device_Type :** Description du capteur connecté à l'entrée analogique.

- **Status_Flags :** cette propriété représente 4 flags (drapeaux) booléens qui indiquent l'état de l'entrée analogique : {IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE} avec :

  **IN_ALARM : a pour valeur « faux » si la propriété « Event_State » a une valeur « NORMAL »

  **FAULT : vaut « TRUE » si la propriété « Reliability » est présente et n'a pas comme valeur «NO_FAULT_DETECTED ».

  **OVERRIDEN: vaut "TRUE" si il y a dépassement: dans ce cas la propriété «Present_Value » ne détecte plus de changements à l'entrée physique.

  ** OUT_OF_SERVICE: a une valeur logique "TRUE" si la propriété « Out_Of_Service » vaut elle aussi « TRUE ».
- **Event_State :** indique l'état de l'objet après un événement.
- **Reliability :** Cette propriété indique si la valeur présente dans la propriété « Present_Value » est fiable ou non. Cette propriété peut avoir les différentes valeurs suivantes : { NO_FAULT_DETECTED, NO_SENSOR, OVER_RANGE, UNDER_RANGE,OPEN_LOOP, SHORTED_LOOP, UNRELIABLE_OTHER}.

Property	Data type	R/W/O	Information
Object_Identifier	*BACnetObjectIdentifier*	R	(Analog Input, Instance 4)
Object_Name	*CharacterString*	R	"Building 007, Room 004"
Object_Type	*BACnetObjectType*	R	ANALOG_INPUT
Present_Value	*Real*	R[1]	25.3
Description	*CharacterString*	O	"Room air temperature"
Device_Type	*CharacterString*	O	"NTC Type 4711"
Status_Flags	*BACnetStatusFlags*	R	{FALSE,FALSE,FALSE,FALSE}
Event_State	*BACnetEventState*	R	NORMAL
Reliability	*BACnetReliability*	O	NO_FAULT_DETECTED
Out_Of_Service	*Boolean*	R	FALSE
Update_Interval	*Unsigned*	O	10 (seconds)
Units	*BACnetEngineeringUnits*	R	°C (measurement unit for Present_ Value)
Min_Pres_Value	*Real*	O	−10.0 (lowest number of the measurement range)
Max_Pres_Value	*Real*	O	60.0 (highest number of the measurement range)
Resolution	*Real*	O	0.2 (smallest recognizable change in Present_Value in units)
COV_Increment	*Real*	O[2]	0.5 (threshold value for transmission/ notification)
Time_Delay	*Unsigned*	O[3]	10 (seconds before error message is sent)
Notification_Class	*Unsigned*	O[3]	5 (Notification Class object Nr. 5)
High_Limit	*Real*	O[3]	40.0 (upper limit)
Low_Limit	*Real*	O[3]	5.0 (lower limit)
Deadband	*Real*	O[3]	2.0 (range between the high and low limits)
Limit_Enable	*BACnetLimitEnable*	O[3]	{TRUE, TRUE} (enable reporting of all high and low limits)
Event_Enable	*BACnetEventTransitionBits*	O[3]	{TRUE, FALSE, TRUE}
Acked_Transitions	*BACnetEventTransitionBits*	O[3]	{TRUE, TRUE, TRUE}
Notify_Type	*BACnetNotifyType*	O[3]	EVENT
Event_Time_ Stamps	*BACnetARRAY[3] of BACnetTimeStamp*	O[3]	((12-JAN-07,10:23:24.3), (*-*-*,*:*:*.*), (26-FEB-07,23:55:22.3)
Profile_Name	*CharacterString*	O	"HBN_Temp_Sensor"

R[1]: is writable when *Out_Of_Service* is TRUE
O[2]: is required when the object supports reporting by minimum change in *Present_Value*
O[3]: is required when the object supports intrinsic reporting

Fig. C.III.19. Exemple de propriétés d'un objet de type ANALOG_INPUT [B8]

- **Out_of_Service :** Cette propriété indique si l'entrée est opérationnelle ou non.
- **Update_Interval :** Cette propriété indique la période de temps maximale entre deux mise à jour de la propriété « Update_Interval »
- **Time_Delay :** cette propriété spécifie le temps minimum pendant lequel « Present_Value » doit rester en dehors des deux limites : « High_Limit » et « Low_Limit » pour qu'un événement de type « TO_OFFNORMAL » est généré.
- **Notification_Class :** spécifie la classe des signalement à utiliser pour cet objet.
- **Deadband :** Si « Present_Value » descend au delà de la valeur « High_Limit » moins « DeadBand » (respectivement prend une valeur au dessus de « Low_Limit » plus « DeadBand ») un événement de type « TO_NORMAL » est généré.
- **Limit_Enable :** Cette propriété permet la génération d'événement lors du dépassement des valeurs limites.
- **Event_Enable :** Cette propriété possède 3 flags (drapeaux) qui permettent ou non la génération des événements : TO_OFFNORMAL, TO_FAULT et TO_NORMAL respectivement.
- **Acked_Transition :** cette propriété concerne les accusé de réception pour : TO_OFFNORMAL, TO_FAULT et TO_NORMAL respectivement.
- **Notify_Type :** indique si la notification générée par l'objet est de type « Event » ou bien « Alarm ».
- **Event_Time_Stamps :** cette propriété représente le temps de la dernière notification pour les evennements : TO_OFFNORMAL, TO_FAULT et TO_NORMAL.
- **Profile_Name :** Propriété additionnelle spécifique au dispositif raccordé à l'entrée analogique.

IV. Conclusion

Durant ce chapitre, nous avons vu les principaux réseaux utilisés dans le monde de la GTB, les réseaux vus s'utilisent à des niveaux différents, en effet, le réseau Modbus et le réseau Lon s'utilisent au niveau de la couche terrain, tandis que le réseau BACnet s'utilise au niveau de couche supervision.

Vue la taille des bâtiments qui ne cesse de grandir et la complexité des taches requise dans un bâtiment intelligent entrainée par le haut niveau de confort exigé, l'utilisation des réseaux en GTB est devenu incontournable aujourd'hui. Ainsi, ce petit annexe donne un aperçue général des réseaux utilisés en GTB qui sera utile pour toutes personne qui veut s'initier dans ce domaine

ANNEXE D

Description de la Technologie Utilisée pour l'Application

I. Matériel utilisé pour l'installation électrique

Dans ce qui suit, nous allons donner un bref descriptif sur les équipements électriques choisis pour notre application.

1. Le disjoncteur Masterpact NW 2500

Le disjoncteur Masterpact de Merlin Gerin possède un pouvoir de coupure pouvant atteindre 150 KA, il dispose d'une unité de contrôle interchangeable communicante.



Fig. D.I.1. Le disjoncteur Masterpact NW 2500 [M3]

1.1. Contacts auxiliaires du Masterpact NW 2500

Le Masterpact NW 2500 possède entre autres les contacts auxiliaires suivants :

- Le contact **Ouvert Fermé OF :** Permettant de signaler l'état du disjoncteur
- Le contact **Signal Défaut SD :** Permettant de signaler un défaut du disjoncteur.



Contact OF **Contact SD**

Fig. D.I.2. Les contacts auxiliaires du Masterpact [M3]

Chacun de ces contacts nécessite trois fils pour le branchement avec l'automate. Pour notre projet, la liaison avec l'automate se fait à travers des modules déportés Momentum.

1.2. Commande du Masterpact NW 2500

La commande du Masterpact se fait via un motoréducteur à courant alternatif, la référence de cet actionneur est le 33179 ; le bornier utilisé pour le branchement est le 47074.

Cet actionneur nécessite trois fils pour le branchement avec un automate.

La tension d'alimentation des moteurs réducteurs varient entre 48V à 480 V pour les moteurs à CA et de 24V à 250V pour les moteurs à CC.

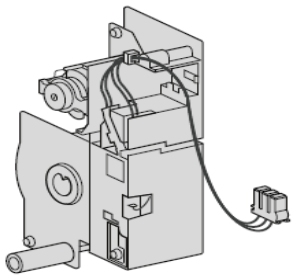


Fig. D.I.3. Le motoréducteur d'un Masterpact [M3]

2. Le disjoncteur Compact NS 1000

Le Compact NS 1000 est un disjoncteur avec pouvoir de sectionnement ayant un courant assigné (nominal) In de 1000A et un pouvoir de coupure de 85 KA, ce disjoncteur possède un déclencheur de type électronique et peut communiquer à travers un bus, ce qui permet la signalisation de l'état de l'appareil et des alarmes.

Le compact NS possède une motorisation qui s'intègre au disjoncteur permettant sa commande à distance.

Grâce à son intelligence Micrologic, le Compact NS peut réaliser divers fonctions de mesure (courant, tension et même puissance), ce qui le dote des fonctions d'un Power Meter.

Ce disjoncteur possède la fonction sectionnement, conformément à la norme IEC 60947-2 et porte en face avant le symbole "disjoncteur sectionneur" 



Fig. D.I .4. Le disjoncteur Compact NS 1000 [M3]

Le disjoncteur est équipé de trois contacts auxiliaires OF pour indiquer son état et d'un auxiliaire SD pour signaler un défaut survenant.

Le compact NS peut être aussi équipé de déclencheurs volumétriques permettant le déclenchement à distance du disjoncteur.

3. Le disjoncteur Compact série NSX

Cette gamme propose des disjoncteurs ayant un courant assigné In allant de 100 à 630A ; pour notre application, nous avons utilisé des disjoncteurs de 100A, 250A et 400A. Le pouvoir de coupure de ces disjoncteurs varie de 36KA à 150KA



Fig. D.I.5. Le disjoncteur Compact NSX [M3]

3.1. Les auxiliaires électriques

Le compact NSX possède un seul modèle d'auxiliaire utilisable sur tous les appareils Compact NSX100 à NSX630, ce modèle réalise 3 fonctions différentes selon l'emplacement dans lequel il est logé (OF, SD, SDE).

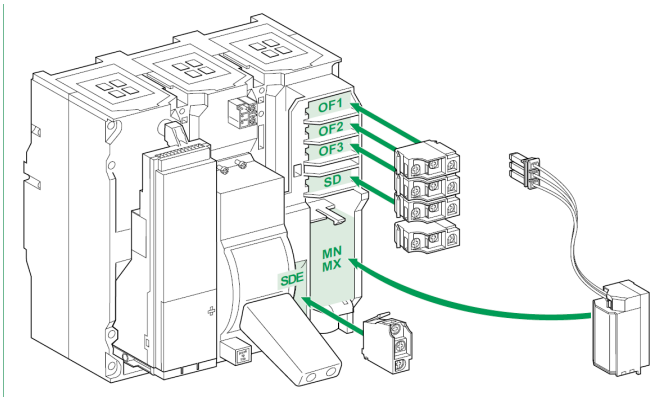


Fig. D.I.6. Emplacement des auxiliaires dans un Compact NSX [M3]

3.2. Télécommande

Les Compact NSX100 à 630 peuvent recevoir un bloc de télécommande adaptable en face avant (la référence est LV432641). Ces blocs sont commandés par une tension alternative de 230 V. Ils restent toutefois actionnables manuellement en face avant.

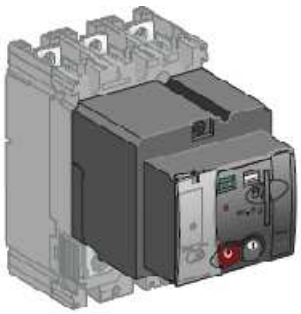


Fig. D.I.7. Bloc de télécommande monté sur le Compact NSX [M3]

4. L'inverseur de source Masterpact 2500 et Masterpact 1000

L'inverseur de source est un élément essentiel pour la continuité de service et la gestion de l'énergie.
Le système réalise la permutation entre :

- une source N qui alimente normalement l'installation
- une source R de remplacement qui peut être une arrivée de réseau supplémentaire ou un groupe électrogène.

L'inverseur de source est réalisé autour de deux appareils interverrouillés mécaniquement. Les interverrouillages empêchent toute mise en parallèle même transitoire des deux sources.

Pour notre application, trois inverseurs de sources ont été prévu pour secourir l'ensemble de l'installation électrique de notre bâtiment (sauf le système de climatisation). Ces inverseurs de sources sont disposés au niveau des TGBT 1 et 2.

4.1. L'inverseur de source Masterpact 2500

L'inverseur de source Masterpact 2500 constitue un choix pour le TGBT 1.
Ce dispositif est réalisé autour de deux disjoncteurs, l'un associé à la source normal et l'autre à la source de secours (le groupe électrogène à travers son TGGE).
Les deux disjoncteurs sont interverrouillés mécaniquement (**Figure D.I.8**) afin d'éliminer tous risque de fermeture simultanée même transitoire.

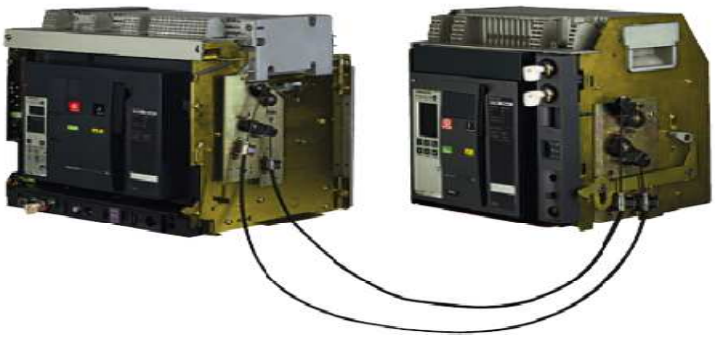


Fig. D.I.8. Interverrouillage par câble de deux disjoncteurs Masterpact constituant un inverseur de source [M3]

L'inverseur de source Masterpact 2500 est aussi équipé d'un interverrouillage électrique. L'interverrouillage électrique s'associe à l'interverrouillage mécanique. Il télécommande le transfert des sources. Ce dispositif peut se compléter par un automatisme qui prend en compte les informations du réseau.

L'interverrouillage électrique est réalisé grâce à un dispositif spécial appelé IVE

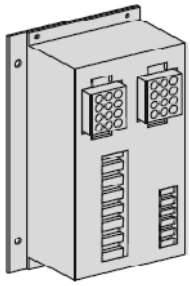


Fig. D.I.9. Dispositif IVE pour interverrouillage électrique [M3]

4.2. L'inverseur de source Masterpact 1000

Cet inverseur de source constitue un choix pour secourir les deux cotés de TGB2, il possède toutes les caractéristiques du masterpact 2500

5. Le disjoncteur série C60N

Les tableaux divisionnaires d'éclairage de notre application contiennent des disjoncteurs de départ permettant la protection.

Le disjoncteur série C60N ayant comme référence 24202 constitue un choix pour notre application. Ce disjoncteur possède un calibre de courant de 16A et un courant de déclenchement de 10 kA.



Fig. D.I.10. Disjoncteur C60N avec bloc différentiel Vigi [M3]

Ce disjoncteur peut être associé à un bloc de différentiel Vigi C60, ce bloc offre une protection contre les déclenchements intempestifs dues aux surtensions passagères (coup de foudre, manœuvre d'appareillage sur le réseau...etc.)

Ce disjoncteur est aussi doté d'un contact auxiliaire OF (la référence est 26924) permettant de signaler son état.

6. Les télérupteurs

L'alimentation des lampes constituant le système d'éclairage de notre application se fait à travers des télérupteurs. Les télérupteurs sont utilisés pour la commande par boutons-poussoirs de circuits d'éclairage constitués de :

- lampes à incandescence, lampes halogènes basse tension, etc. (récepteurs résistifs)
- tubes fluorescents, lampes à décharge, etc. (récepteurs inductifs)

Le télérupteur ayant comme référence 15032 de Schneider constitue un choix pour notre application. Il possède un calibre de courant de 16A, la tension de commande est de 230 V CA.



Fig. D.I.11. Le télérupteur 15032 de Schneider [M4]

La figure ci-dessus illustre le fonctionnement du télérupteur, lorsque celui-ci reçoit une impulsion de commande de l'automate (schématiser par un bouton poussoir dans la figure), la bobine du télérupteur se trouvera branchée entre la phase et le neutre et elle sera donc excitée, ceci engendrera la fermeture du contact alimentant le circuit d'éclairage et donc l'allumage de ce dernier.

II. Matériel utilisé pour le système d'éclairage

1. Les ballasts

1.1. Introduction et fonctionnement du ballast

Le fonctionnement des lampes fluorescentes et des lampes à décharge nécessite l'utilisation de ballasts et de starters (pour les lampes fluo) ou d'amorceurs (pour les lampes à décharge).

L'exemple repris ici décrit le mode d'allumage d'un tube fluorescent. Le fonctionnement est identique pour les lampes fluocompactes et pour les lampes à décharge. Pour ces dernières, le brûleur remplace le tube et l'amorceur remplace le starter.

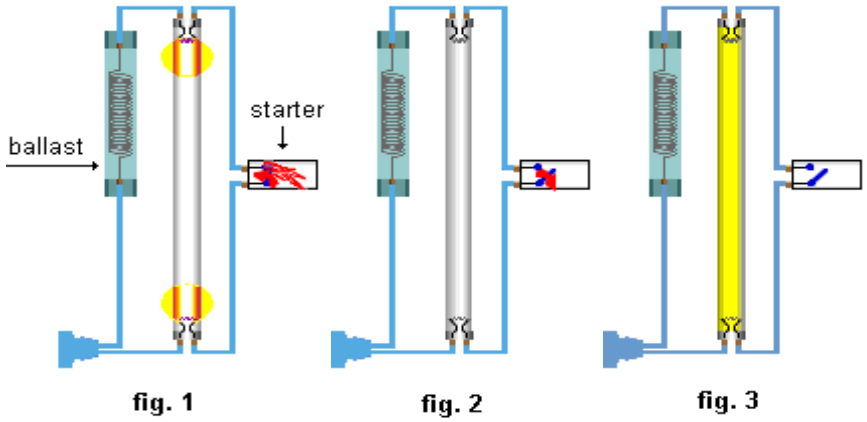


Fig. D.II.1. Emplacement des ballasts et fonctionnement du starter [W6]

Le starter est composé d'un petit tube rempli de gaz et pourvu d'un bilame.

A l'allumage, la mise sous tension provoque un arc électrique au sein du gaz. Celui-ci échauffe le bilame, jusqu'alors ouvert *(fig. 1)*, qui se ferme.

Pendant ce temps, un courant circule dans les électrodes. Elles s'échauffent et ionisent le gaz qui les environne, ce qui facilitera l'allumage.

Le bilame étant fermé, l'arc électrique dans le starter disparaît.

Le bilame se refroidit alors et s'ouvre *(fig. 2)*. Il provoque ainsi une interruption brusque du courant dans le ballast raccordé en série.

Le **ballast**, composé d'un bobinage de cuivre entourant un noyau de fer (ballast dit inductif ou électromagnétique), va tenter de rétablir ce courant en libérant toute son énergie. Cela provoque une impulsion de tension très élevée entre les électrodes de la lampe (jusqu'à 1 500 V) capable d'allumer le tube fluorescent *(fig. 3)*.

Souvent, cet allumage ne réussit pas en une seule tentative. Si la lampe ne s'est pas allumée, le cycle recommence.

En fonctionnement, la tension aux bornes de la lampe est trop faible pour générer un nouveau cycle d'allumage (40 à 110 V). Le starter se maintient donc en position ouverte et le courant traverse la lampe qui reste allumée.

A partir de cet instant, le ballast joue le rôle de limiteur de courant et empêche la destruction de la lampe.

Lorsque le ballast est électromagnétique, il faudra ajouter un condensateur dans le circuit pour compenser le $\cos(\varphi)$ inductif.

Il existe deux types de ballast ; le ballast électromagnétique et le ballast électronique, dans ce qui suit, nous allons exposer leurs propriétés brièvement

1.2. Types de ballast

1.2.3. Le Ballast électromagnétique

Le ballast électromagnétique (appelé aussi "inductif" ou "conventionnel") est essentiellement constitué d'un bobinage. Il doit être associé à un starter pour provoquer l'allumage des lampes fluorescentes.. Certains ballasts dits "à faibles pertes", ont une consommation nettement plus faible que celle des ballasts conventionnels. Il existe aussi des ballasts "à très faibles pertes" mais ils sont beaucoup plus volumineux.

L'utilisation de ballasts électromagnétiques induit un facteur de puissance relativement bas (cos(φ 0,5), ce qui en tarif Haute-Tension est pénalisé par le distributeur électrique. Il n'est donc pas rare de devoir ajouter des condensateurs soit en tête d'installation, soit au niveau des luminaires afin de compenser l'effet inductif. On peut aussi insérer des condensateurs (d'une capacité double) dans 50 % des circuits de lampes pour compenser l'effet inductif total. Ceci permet d'économiser un condensateur sur deux.

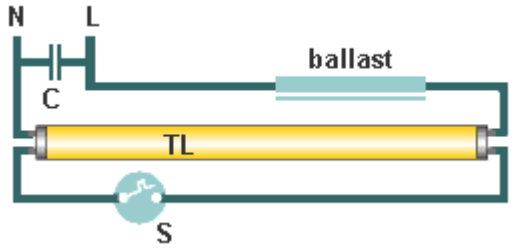


Fig. D.II.2. Correction du facteur de puissance pour un ballast électromagnétique [W6]

1.2.4. Ballast électronique

L'ensemble starter, ballast conventionnel et condensateur de compensation du cos (φ) peut être remplacé par un ballast électronique avec ou sans préchauffage des cathodes.
Celui-ci alimente les lampes sous haute fréquence (entre 25 et 60 kHz). Il est appelé également ballast HF (haute fréquence).
Son facteur de puissance est proche de 1 et il n'y a donc pas de nécessité de compenser celui-ci par l'utilisation de condensateurs.
La particularité du ballast électronique est sa capacité à faire varier la tension de l'arc électrique dans une lampe fluorescente, ce qui permet de la dimmer. Ainsi, ce ballast constitue l'équipement adéquat pour une application d'éclairage en fonction de la lumière du jour nécessitant la variation de l'intensité d'éclairage des lampes (*dimming*)

Un ballast électronique est constitué des éléments suivants :

- Un étage redresseur
- Un condensateur de filtrage
- Un étage onduleur en demi pont
- un étage adaptateur tension/courant, soit régulé par miroir de courant ou pour les modèles les plus perfectionné, par microcontrolleur

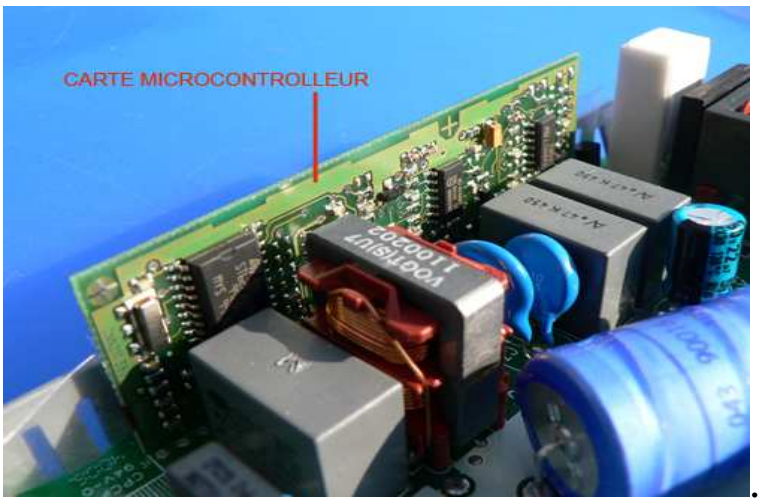


Fig. D.II.3. Constituant d'un ballast électronique [W6]

2. Le capteur de luminosité de jour

Ce capteur est à utilisé pour le mode de la gestion d'éclairage en fonction de la luminosité du jour.

Le capteur SLR310 fabriqué par la firme TAC (filière de Schneider) constitue un choix pour notre application.

2.1. Caractéristique du capteur de luminosité SLR 310

- température de fonctionnement -20… +70 °C
- précision ±5 à 25 °C
- plage de mesure (choix par cavaliers) 0 - 400 lx ou 0 - 20 klx
- signal de sortie 0 - 10 V (3 fils)
- dimensions 75 x 75 x 25 mm
- alimentation 24 V CA ou 15 V CC
- montage mural



Fig. D.II.4. Le capteur de luminosité SLR 310 [M5]

3. Capteur de mouvement

Le capteur de mouvement à été utilisé dans notre application pour éteindre l'éclairage dans certain bureaux lorsque le capteur indique qu'il n'ya aucune présence.

Le capteur PIR-TF-25 de TAC Automation constitue un choix pour notre application.



Fig. D.II.5. Le capteur de luminosité PIR-TF-25 [M5]

3.1. Caractéristiques du capteur PIR-TF-25

- température de fonctionnement -20… +50 °C
- hauteur de montage 1,5 à 3 m
- alimentation 24 V CA / CC
- consommation 18 mA
- signal de sortie : 0-10V
- intervalle de détection choix par cavaliers (angle de détection pouvant atteindre 180°)

4. Interrupteur Crépusculaire

Pour la gestion du système d'éclairage extérieur de notre application, on a prévu des interrupteurs crépusculaires à seuil réglable. L'interrupteur crépusculaire LUNA 111 constitue un choix pour notre application. Il possède les caractéristiques suivantes :

- Alimentation : 230-240 V 50 à 60 Hz +10/-15 %
- Consommation : 3 VA
- Seuil de luminosité réglable de 1 à 99 klux
- Temporisation ON ou OFF réglable de 0 à 59 minutes
- Longueur de câble sur entrées externes : maximum 100 m
- Contacts : 1 ou 2 inverseurs

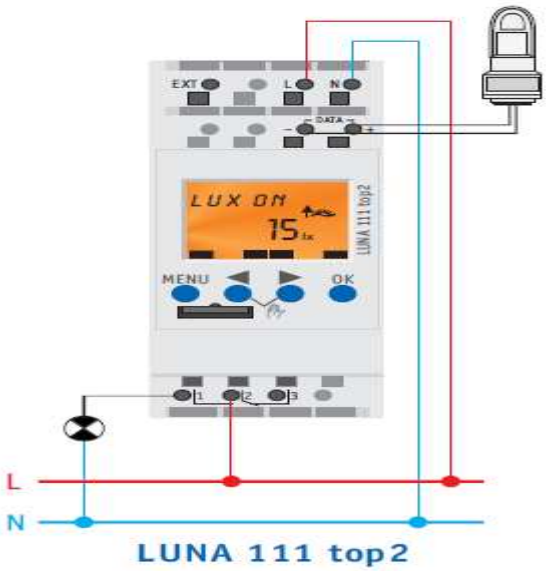


Fig. D.II.6. Interrupteur crépusculaire

5. Interrupteur automatique infrarouge WK051 Kallysta

Cet élément n'a pas été utilisé dans notre application, on le présente ici juste pour montrer les tendances de matériel utilisé en GTB.

L'interrupteur WK051 est un composant communicant (réseau KNX) multifonction, permettant une commande de l'éclairage en fonction de la présence ou encore de la lumière du jour. Pour cela il possède différents capteurs intégrés.

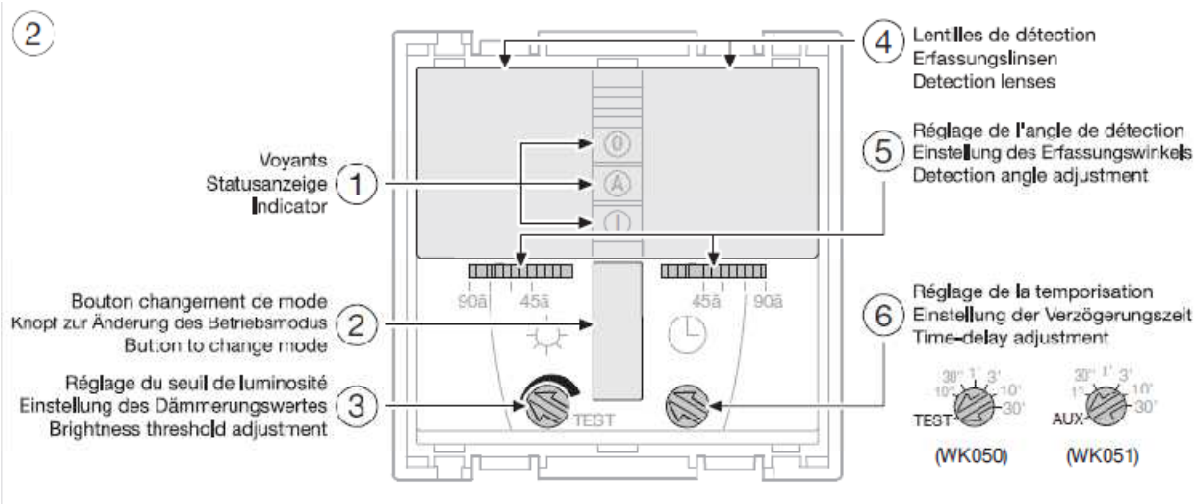


Fig. D.II.7 Fonction de l'interrupteur automatique infrarouge WK051 [F9]

Caractéristiques techniques	Valeurs
Angle de détection	180 °
Tension nominale	230 V
Plage de temporisation	1...30min
Sans halogène	Oui
Série	kallysta

Tableau D.II.1. Caractéristiques de l'interrupteur Automatique [F9]

III. Matériel utilisé pour la CTA (Centrale de Traitement d'Air)

1. Les capteurs

1.1. Les capteurs analogiques

1.1.1 Sonde de température d'air neuf « TT 531 » [F4]

Cette sonde mesure la température de l'air venant de l'extérieur (air neuf).Le capteur est monté dans un bouchon en laiton afin d'éviter la perturbation issue de l'ensoleillement direct.



Fig. D.III.1. Sonde de température TT531 [F4]

Les différents modèles disponibles pour cette sonde sont résumés sur le tableau suivant :

Référence	Capteur T°	Type	*
TT531/20K6A1	Sonde Thermistance	CTN 20K6	
TT531/0-10 Vdc/1	Sonde Platine avec transmetteur	0-10 Vdc –10°C à +40°C	
TT531/0-10 Vdc/3	Sonde Platine avec transmetteur	0-10 Vdc –10°C à +110 °C	
TT531/4-20mA/1	Sonde Platine avec transmetteur	4-20 mA –10°C à +40°C	
TT531/4-20mA/3	Sonde Platine avec transmetteur	4-20 mA –10°C à +110 °C	
TT531/PT1000A	Sonde Platine	PT1000	

Tableau. D.III.1. Les différents modèles de la sonde de température TT531 [F4]

Les caractéristiques de ce capteur sont résumées dans le tableau suivant :

Type de capteur de température	Pt1000 A capteur platine 1000 ohms à 0°C 20 K6 AI capteur CTN Pt100 A capteur Pt100 incorporé modèles avec transmetteur 0-10 Vdc ou modèle 4-20 mA
Précision	Thermistance +/- 0.2 K de 0°C à 70°C Pt100 +/- 0.35 K de 0°C à 100°C Pt1000 +/- 0.35 K de 0°C à 100°C
Réceptacle du capteur	Matériau en laiton
Boitier	Matériau : ABS non propagateur de flamme Protection IP 65
Conditions ambiantes	-10°C à +60°C
Alimentation (modèles avec transmetteur)	24Vdc +/- 10%
Sorties (selon le modèle)	Sortie directe : Résistance de l'élément CTN Sortie 0 –10 Vdc ou 4 –20 mA
Démentions	Boîtier : 85 mm x 85mm x 37 mm

Tableau. D.III.2. Caractéristiques de la sonde de température TT531 [F4]

La carte électronique de la **version 0-10V** (que nous avons choisi pour notre application) est illustrée à la figure suivante **:**

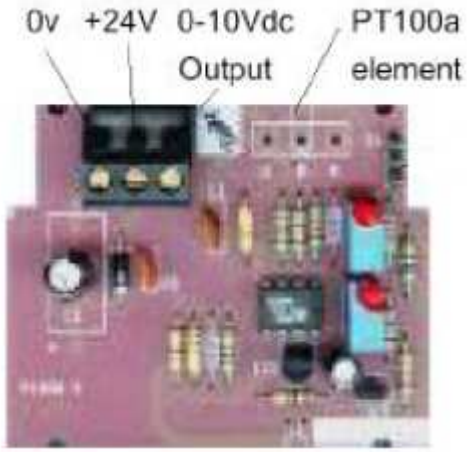


Fig. D.III.2. Carte électronique de la sonde de température TT531 [F4]

1.1.2 Sonde de température d'air soufflé et repris : sonde de gaine « KTM1 » [F5]

La sonde de température pour gaine d'air **THERMASGARD KTM1** avec convertisseur et 8 plages de mesure commutable à sortie analogique s'insère facilement dans les gaines de ventilation et de climatisation.



Fig. D.III.3. Sonde de température KTM1 [F5]

Les caractéristiques techniques de ce capteur sont résumées dans le tableau suivant :

Tension d'alimentation	24V ca/cc ±10% pour sortie 0-10V 15-35V cc pour sortie 4...20mA (dépend de la charge)
Plages de mesure	**commutation multi-gamme** **avec 8 plages de mesure commutables**
Sortie	0-10V ou 4...20mA
Température ambiante	-30...+70°C
Type de raccordement	2 ou 3 fils
Démentions	72 x 64 x 39,4mm
Raccordement électrique	0,14-1,5mm², par bornes à vis sur carte
Raccordement au process	par bride (compris dans la livraison), matière plastique (acier zingué en option, voir accessoire)
Humidité d'air admissible	<95% h.r., sans condensation de l'air
En option	écran avec rétro-éclairage, affichage monoligne, découpe 37x15mm (lxh), pour afficher la température effective

Tableau. E.III.3. Caractéristiques de la sonde de température KTM1 [F5]

Les differentes gammes de la sonde de température ETM1 sont illustrées dans le tableau suivant :

Désignation/WG1	sortie	50mm	100mm	150mm	200mm	250mm	300mm
KTM1-I	4...20mA	•	•	•	•	•	•
KTM1-U	0-10V	•	•	•	•	•	•
KTM1xx-Display	écran avec rétro-éclairage, affichage monoligne						
En option:	d'autres plages de mesure sur demande						
Pour commandes veuillez indiquer:	longueur du tube de protection par ex. KTM1-U, 150mm; KTM1-I, 300mm						

Tableau. D.III.4. Les differentes gammes de la sonde de température KTM1 [F5]

Nous avons choisi de prendre comme sonde la **KTM1-U sortie 0-10V (3 fils)** dont voici les conditions de raccordement électrique :

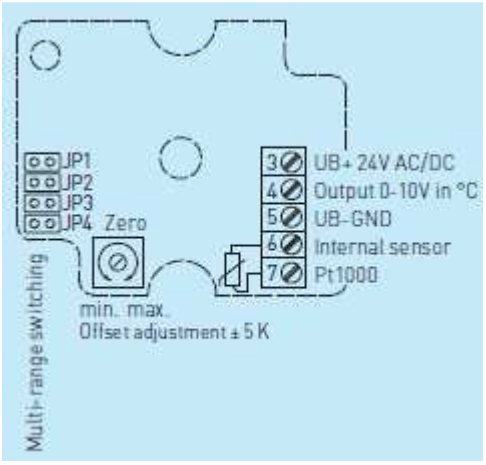


Fig. D.III.4. Raccordement electrique de la sonde KTM1-U [F5]

Les differentes plages de mesure sont illustrées quand à elles dans le tableau suivant :

Plages de mesure (réglables)	JP 1	JP 2	JP 3	JP 4
-20 °C ... +150 °C	●	●	●	●
-50 °C ... +50 °C	○ ○	●	●	●
-20 °C ... +80 °C	●	○ ○	●	●
-30 °C ... +60 °C	○ ○	○ ○	●	●
0 °C ... +40 °C	●	●	○ ○	●
0 °C ... +50 °C	○ ○	●	○ ○	●
0 °C ... +100 °C	●	○ ○	○ ○	●
0 °C ... +150 °C	○ ○	○ ○	○ ○	●

Tableau. D.III.5. Les différentes plages de mesure de la sonde de température KTM1-U [F5]

Le schéma de montage de cette sonde à la gaine d'air est illustré par la figure suivante :

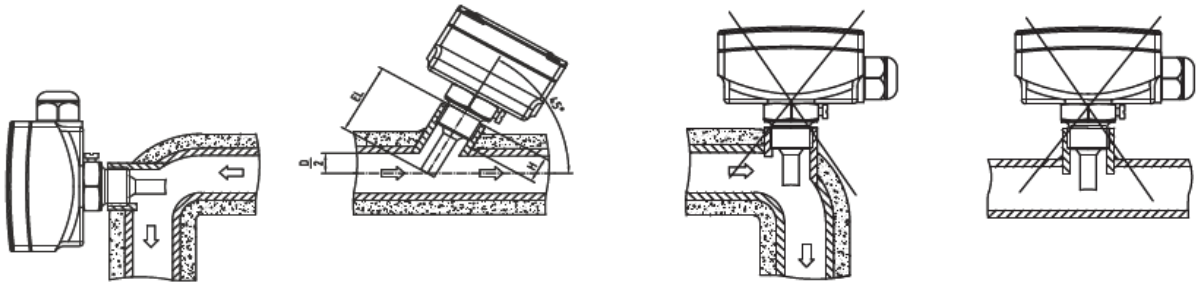


Fig. D.III.5. Montage de la sonde KTM1-U dans une gaine d'air [F5]

1.2. Les capteurs digitaux

1.2.1 Pressostat Différentiel « DPS » [F6]

Les pressostats différentiels à air servent à contrôler la différence de pression en amont et en aval des filtres à air. L'un des grands avantages de ce DPS (Differential Pressure Switches) est sans doute son point de commutation facilement réglable avec échelle en Pascal.

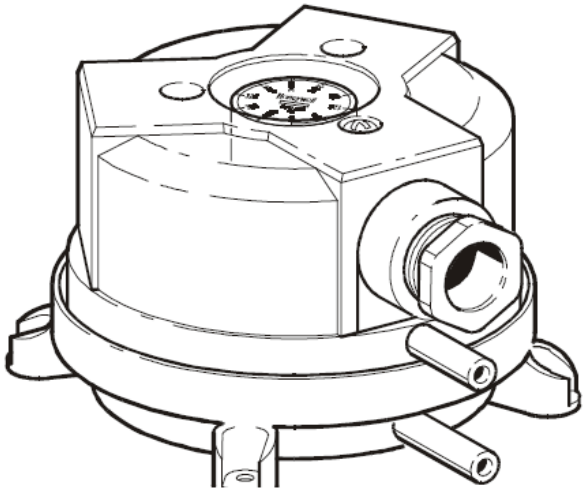


Fig. D.III.6. Présostat différentiel DPS [F6]

Les différentes gammes disponibles sont illustrées par le tableau suivant :

Modèle	Echelle de réglage de à	Différentiel de commutation (hystérésis)
DPS 200	20 – 200 (Pa)	10 (Pa)
DPS 400	40 – 400 (Pa)	20 (Pa)
DPS 1000	200 – 1000 (Pa)	100 (Pa)
DPS 2500	500 – 2500 (Pa)	150 (Pa)

Tableau. D.III.6. Les différentes gammes du pressostat différentiel [F6]

Les caractéristiques de ce DPS sont résumées dans le tableau suivant :

Gaz Contrôlé	Air, gaz non inflammable et non agressif.
Pression max. de fonctionnement	5000 Pa
Raccords de prise de pression	2 tubes de plastique ø 6mm extérieur
Pattes de fixation	Incluses dans le boitier
Conditions de fonctionnement	-20°C à 85°C

Membrane	Silicone
Alimentation	AC
Type de sortie	Sortie contact AC

Tableau. D.III.7. Caractéristiques du pressostat différentiel [F6]

Le raccordement de ce pressostat est illustré à la figure suivante :

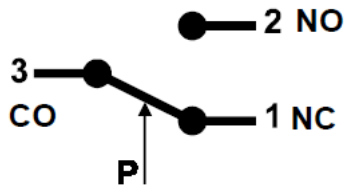


Fig. D.III.7. Raccordement du Présostat différentiel DPS [F6]

2. Les Actionneurs

2.1. Les actionneurs analogiques

2.1.1 Vanne mélangeuse 3 voies

Pour alimenter la batterie froide avec de l'eau à basse température (température modulable), on réalise un mélange entre l'eau froide qui arrive du groupe frigorifique (**GEG** : Groupe Eau Glacée) et l'eau qui arrive de la batterie froide et qui retourne vers le GEG.

C'est le rôle de la vanne, appelée "vanne trois voies mélangeuse", placée entre l'aller et le retour de l'installation qui permet d'effectuer ce mélange. Son principe de fonctionnement est basé sur la rotation d'un secteur entre les 3 voies d'eau comme le montre la figure suivante :

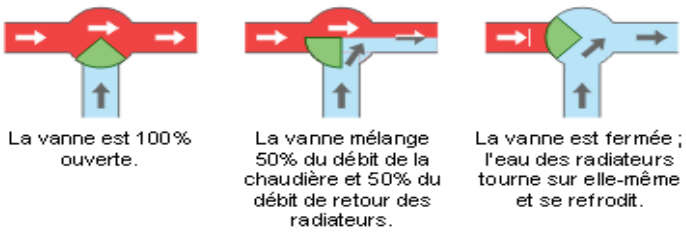


Fig. D.III.8. Principe de fonctionnement de la vanne 3 voies mélangeuse [W6]

Nous avons pris comme exemple les vannes 3 voies de la gamme **karno** dont les références sont résumées dans le tableau suivant :

Référence	Désignation
V3D20-M23	Vanne 3 voies DN 20 Moteur 230V 3 Points
V3D20-M49	Vanne 3 voies DN 20 Moteur 0...10V
V3D25-M23	Vanne 3 voies DN 25 Moteur 230V 3 Points
V3D25-M49	Vanne 3 voies DN 25 Moteur 0...10V
V3D32-M23	Vanne 3 voies DN 32 Moteur 230V 3 Points
V3D32-M49	Vanne 3 voies DN 32 Moteur 0...10V
V3D40-M23	Vanne 3 voies DN 40 Moteur 230V 3 Points
V3D40-M49	Vanne 3 voies DN 40 Moteur 0...10V
V3D50-M23	Vanne 3 voies DN 50 Moteur 230V 3 Points
V3D50-M49	Vanne 3 voies DN 50 Moteur 0...10V

Tableau. D.III.8. Les différentes gammes de vannes 3 voies « karno » [F7]



Fig. D.III.9. Exemple d'une vanne 3 voies « karno » [F7]

Ces vannes sont actionnées par des servomoteurs dont voici les principales caractéristiques :

	M23	**M49**
Tension d'alimentation	230 VAC +6% -10%	24 VAC ± 10%
Consommation	7 VA	3 VA
Signal de télécommande	3-points	0 .. 10 Vdc (70 kOhm)

Tableau. E.III.9. Les principales caractéristiques des servomoteurs pilotant les vannes 3 voies « karno » [F7]

Pour notre application nous avons choisi de prendre une vanne 3 voies pilotée par un moteur 0-10V (**2fils**).

IV. Momentum utilisé pour notre application

Les références et les caractéristiques des embases pour Momentum utilisés pour notre application sont les suivantes :



170 ADI ●●0 ●0

Type of current	Input voltage	Modularity (no. of points)	Conformity EC 1131-2	Reference	Weight kg
DC	24 V	16 (1 x 16)	Type 1	170 ADI 340 00	0.190
		32 (2 x 16)	Type 1	170 ADI 350 00	0.200
AC	100…120 V	16 (2 x 8)	Type 2	170 ADI 540 50	0.284
	200…240 V	16 (2 x 8)	Type 2	170 ADI 740 50	0.284



170 ADO ●●0 ●0

Type of current	Output voltage	Modularity (no. of points)	Current per output	Reference	Weight kg
DC solid state, protected	24 V	16 (2 x 8)	0.5 A	170 ADO 340 00	0.210
		32 (2 x 16)	0.5 A	170 ADO 350 00	0.210
AC triac, protected, 1 fuse per group	100…120 V	8 (2 x 4)	2 A	170 ADO 530 50	0.320
		16 (2 x 8)	0.5 A	170 ADO 540 50	0.284
	200…240 V	8 (2 x 4)	2 A	170 ADO 730 50	0.320
		16 (2 x 8)	0.5 A	170 ADO 740 50	0.284



Type of inputs	Number of channels	Ranges	Reference	Weight kg
12 bits + sign	16	± 5 V, ± 10 V, 4-20 mA	170 AAI 140 00	0.215
15 bits + sign	4, differential	Pt 100, Pt 1000, Ni 100 thermocouples B, E, J, K, N, R, S, T	170 AAI 520 40	0.215
	8, differential	± 5 V, ± 10 V, 1-5 V± 20 mA, 4-20 mA	170 AAI 030 00	0.215



170 AAO ●2● 00

Analog output bases

Type of inputs/ outputs	Number of channels	Ranges	Reference	Weight kg
12 bits + sign	4	± 10 V, 0-20 mA	170 AAO 120 00	0.215
		± 10 V, 4-20 mA	170 AAO 921 00	0.215

Fig. D.IV.1. Caractéristiques des Momentum utilisés [F8]

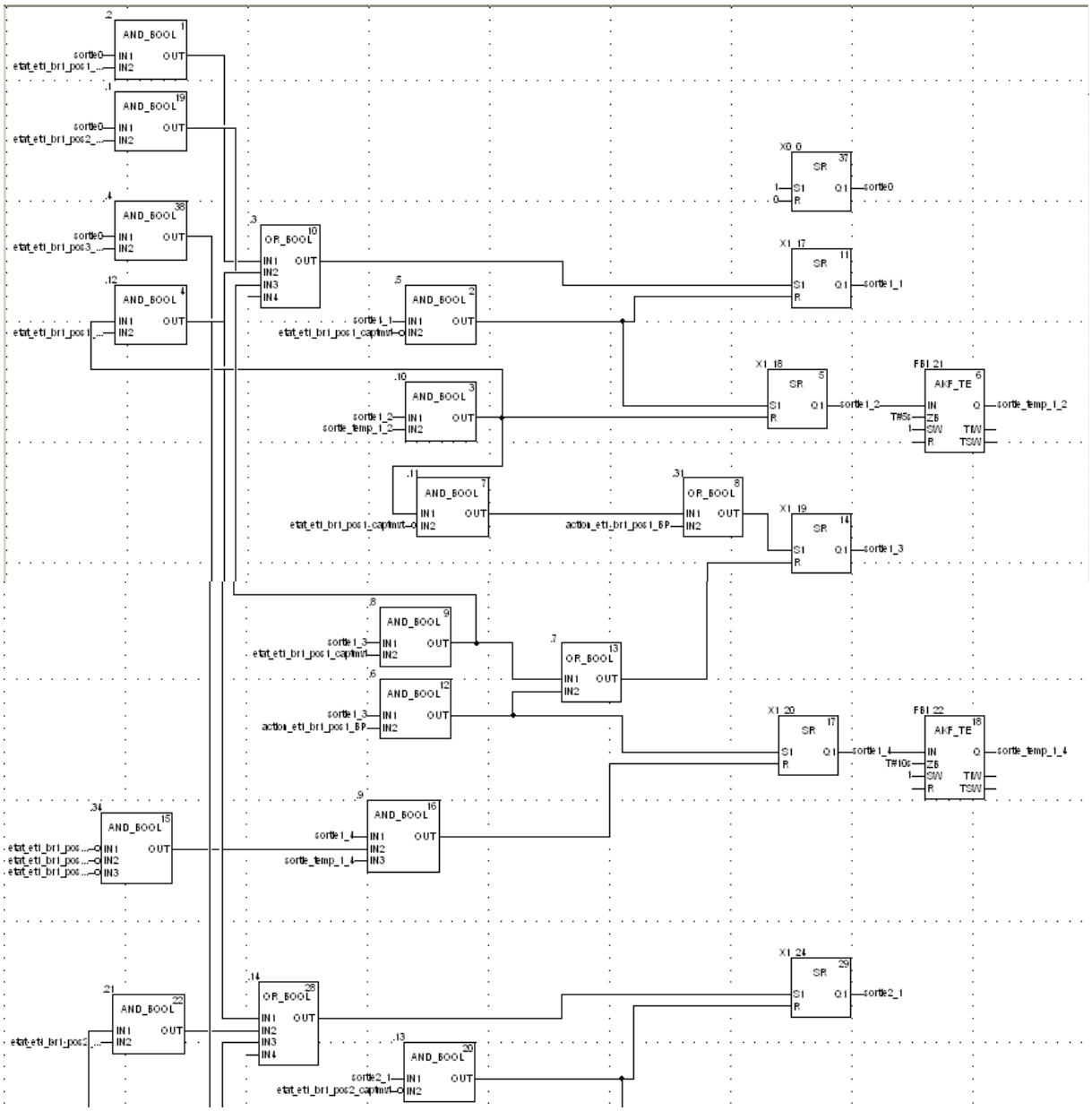
ANNEXE E

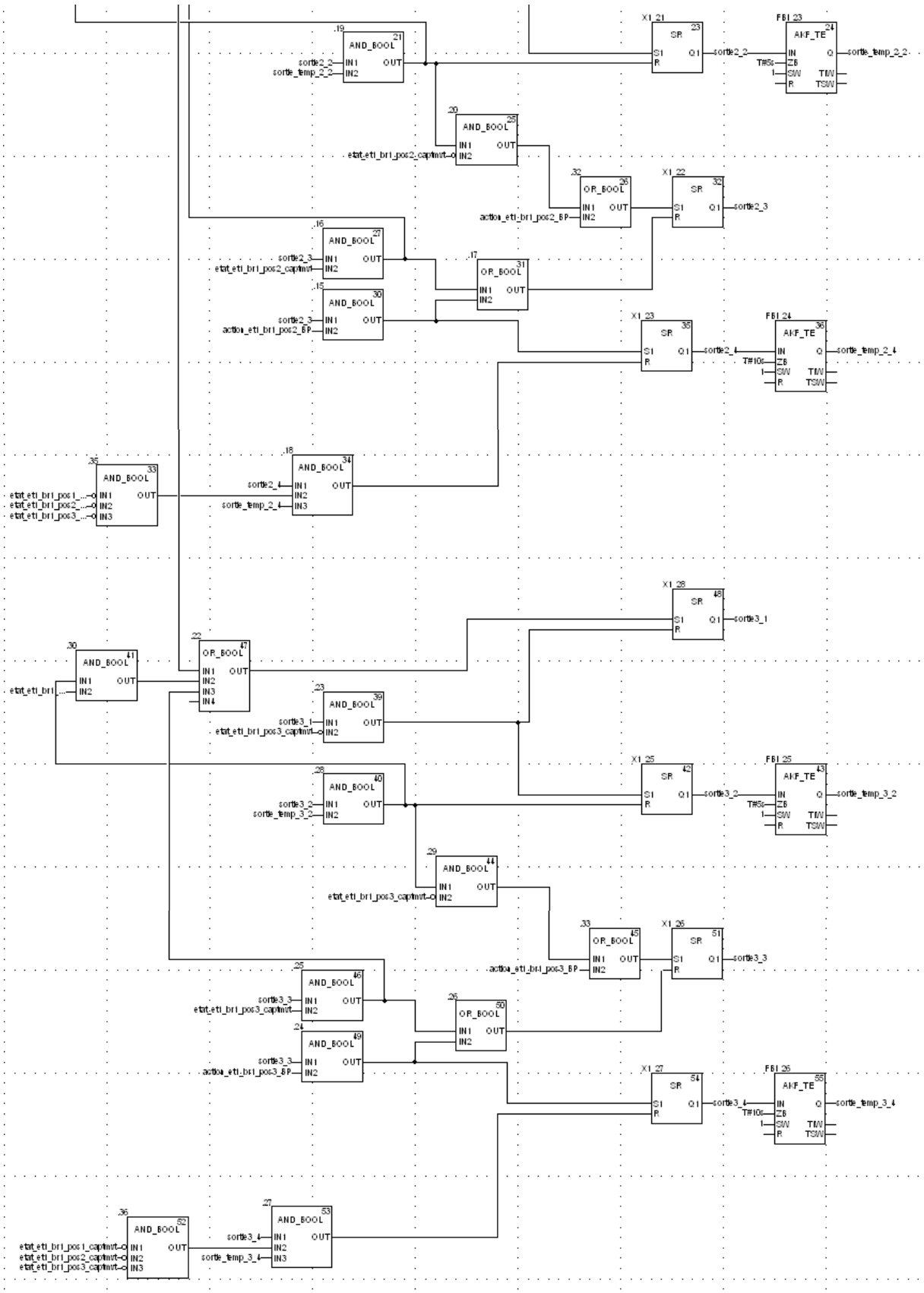
Programmes des Exemples et de l'Application

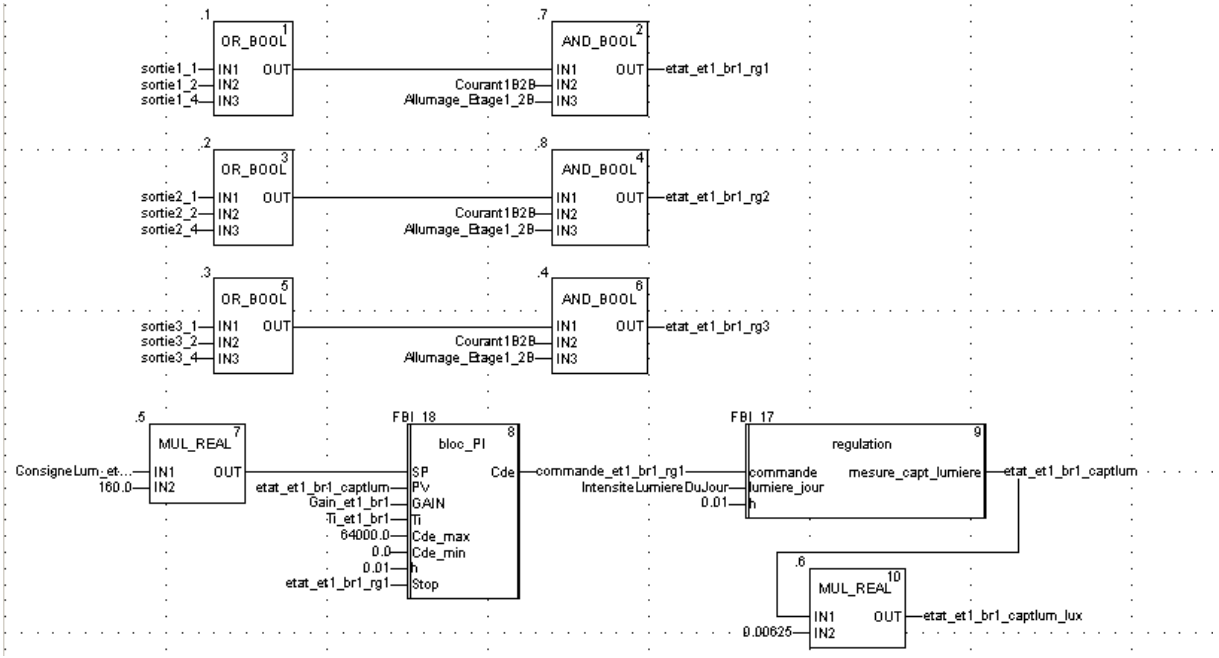
1. Programmes sous Unity Pro

1.1 Programme de l'exemple de l'allumage des luminaires dans un bureau

Les programme en FBD est donné par la figure suivante :




AND_BOOL
OR_BOOL
SR
ANF_TE
sortie2_2
sortie_temp_2_2
sortie2_3
sortie2_4
sortie_temp_2_4
sortie3_1
sortie3_2
sortie_temp_3_2
sortie3_3
sortie3_4
sortie_temp_3_4
etat_eti_bri_pos2_captmvt
action_eti_bri_pos2_BP
etat_eti_bri_pos3_captmvt
action_eti_bri_pos3_BP
T#5s
T#10s



1.2 Programme de gestion et de simulation de l'installation éléctrique

```
(*Actualisation des états des disjoncteurs*)
EtatGEG_Disjoncteur:=ActionGEG_Disjoncteur;
EtatCTA1_Disjoncteur:=ActionCTA1_Disjoncteur;
EtatCTA2_Disjoncteur:=ActionCTA2_Disjoncteur;
EtatCTA3_Disjoncteur:=ActionCTA3_Disjoncteur;
EtatCTA4_Disjoncteur:=ActionCTA4_Disjoncteur;
EtatOnduleur_Disjoncteur:=ActionOnduleur_Disjoncteur;
EtatASC1_Disjoncteur:=ActionASC1_Disjoncteur;
EtatASC2_Disjoncteur:=ActionASC2_Disjoncteur;
EtatASC3_Disjoncteur:=ActionASC3_Disjoncteur;
EtatPompe1_Disjoncteur:=ActionPompe1_Disjoncteur;
EtatTDA1_Disjoncteur:=ActionTDA1_Disjoncteur;
EtatTDA2_Disjoncteur:=ActionTDA2_Disjoncteur;
EtatTDA3_Disjoncteur:=ActionTDA3_Disjoncteur;
EtatTDA4_Disjoncteur:=ActionTDA4_Disjoncteur;
EtatTDA5_Disjoncteur:=ActionTDA5_Disjoncteur;
EtatTDA6_Disjoncteur:=ActionTDA6_Disjoncteur;
EtatTD_SSA_Disjoncteur:=ActionTD_SSA_Disjoncteur;
EtatTD_RDCA_Disjoncteur:=ActionTD_RDCA_Disjoncteur;
EtatTDB1_Disjoncteur:=ActionTDB1_Disjoncteur;
EtatTDB2_Disjoncteur:=ActionTDB2_Disjoncteur;
EtatTDB3_Disjoncteur:=ActionTDB3_Disjoncteur;
EtatTDB4_Disjoncteur:=ActionTDB4_Disjoncteur;
EtatTDB5_Disjoncteur:=ActionTDB5_Disjoncteur;
EtatTDB6_Disjoncteur:=ActionTDB6_Disjoncteur;
EtatTD_SSB_Disjoncteur:=ActionTD_SSB_Disjoncteur;
EtatCCOA1_Disjoncteur:=ActionCCOA1_Disjoncteur;
EtatCCOA2_Disjoncteur:=ActionCCOA2_Disjoncteur;
```

```
EtatCCOA3_Disjoncteur:=ActionCCOA3_Disjoncteur;
EtatCCOA4_Disjoncteur:=ActionCCOA4_Disjoncteur;
EtatCCOA5_Disjoncteur:=ActionCCOA5_Disjoncteur;
EtatCCOA6_Disjoncteur:=ActionCCOA6_Disjoncteur;
EtatCCOB1_Disjoncteur:=ActionCCOB1_Disjoncteur;
EtatCCOB2_Disjoncteur:=ActionCCOB2_Disjoncteur;
EtatCCOB3_Disjoncteur:=ActionCCOB3_Disjoncteur;
EtatCCOB4_Disjoncteur:=ActionCCOB4_Disjoncteur;
EtatCCOB5_Disjoncteur:=ActionCCOB5_Disjoncteur;
EtatCCOB6_Disjoncteur:=ActionCCOB6_Disjoncteur;
Etat1A_Disjoncteur:=Action1A_Disjoncteur;
Etat1B_Disjoncteur:=Action1B_Disjoncteur;
EtatGE1_Disjoncteur:=ActionGE1_Disjoncteur;
EtatGE4_Disjoncteur:=ActionGE4_Disjoncteur;
EtatGE5_Disjoncteur:=ActionGE5_Disjoncteur;
EtatGE6_Disjoncteur:=ActionGE6_Disjoncteur;
EtatCouplage1_Disjoncteur:=ActionCouplage1_Disjoncteur;
EtatExt_Disjoncteur:=ActionExt_Disjoncteur;
EtatTD_Ext_Disjoncteur:=ActionTD_Ext_Disjoncteur;
Etat2A_Disjoncteur:=Action2A_Disjoncteur;
Etat2B_Disjoncteur:=Action2B_Disjoncteur;
Etat3A_Disjoncteur:=Action3A_Disjoncteur;
Etat3B_Disjoncteur:=Action3B_Disjoncteur;
Etat4A_Disjoncteur:=Action4A_Disjoncteur;
Etat4B_Disjoncteur:=Action4B_Disjoncteur;
Etat5A_Disjoncteur:=Action5A_Disjoncteur;
Etat5B_Disjoncteur:=Action5B_Disjoncteur;
Etat6A_Disjoncteur:=Action6A_Disjoncteur;
Etat6B_Disjoncteur:=Action6B_Disjoncteur;
EtatRDCA_Disjoncteur:=ActionRDCA_Disjoncteur;
EtatSSA_Disjoncteur:=ActionSSA_Disjoncteur;
EtatSSB_Disjoncteur:=ActionSSB_Disjoncteur;
EtatTGBT1_Disjoncteur:=ActionTGBT1_Disjoncteur;
EtatTGCO1_Disjoncteur:=ActionTGCO1_Disjoncteur;

(*poste_livraison*)
CouleurAR1_Disjoncteur:=etat_reseau1;
CouleurAR2_Disjoncteur:=etat_reseau2;
EtatAR1_Disjoncteur:=etat_reseau1;
EtatAR2_Disjoncteur:=AND(etat_reseau2,NOT(etat_reseau1));
IF EtatCouplage1_Disjoncteur=0 THEN
CouleurTransfo1_Disjoncteur:=AND(EtatAR1_Disjoncteur,CouleurAR1_Disjoncteur);
CouleurTransfo3_Disjoncteur:=AND(EtatAR2_Disjoncteur,CouleurAR2_Disjoncteur);
  ELSIF EtatCouplage1_Disjoncteur=1 THEN
          IF etat_reseau1=0 THEN
          CouleurTransfo3_Disjoncteur:=AND(EtatAR2_Disjoncteur,CouleurAR2_Disjoncteur);
          CouleurTransfo1_Disjoncteur:=CouleurTransfo3_Disjoncteur;

       ELSIF etat_reseau1=1 THEN
     CouleurTransfo1_Disjoncteur:=AND(EtatAR1_Disjoncteur,CouleurAR1_Disjoncteur);
     CouleurTransfo3_Disjoncteur:=CouleurTransfo1_Disjoncteur;
     END_IF;
```

```
END_IF;
CouleurTransfo2_Disjoncteur:=CouleurTransfo1_Disjoncteur;
CouleurTransfo4_Disjoncteur:=CouleurTransfo3_Disjoncteur;
CouleurCouplage1_Disjoncteur:=CouleurTransfo1_Disjoncteur;

(*TGBT1*)
CouleurTGBT1_Disjoncteur:=AND(EtatTransfo1_Disjoncteur,CouleurTransfo1_Disjoncteur);
Couleur1Inv1_Selecteur:=AND(EtatTransfo2_Disjoncteur,CouleurTransfo2_Disjoncteur);
Couleur2Inv1_Selecteur:=AND(EtatGE4_Disjoncteur,CouleurGE4_Disjoncteur);
EtatInv1_Selecteur:=OR(etat_reseau1,etat_reseau2);
intermediaire1:=AND(EtatTGBT1_Disjoncteur,CouleurTGBT1_Disjoncteur);
intermediaire2:=OR(Couleur1Inv1_Selecteur,Couleur2Inv1_Selecteur);
intermediaire3:=OR(etat_reseau1,etat_reseau2);
IF (intermediaire3=1)AND((intermediaire1=0)OR(intermediaire2=0)) THEN
        EtatCouplage2_Disjoncteur:=1;
   ELSE
        EtatCouplage2_Disjoncteur:=0;
END_IF;
IF EtatCouplage2_Disjoncteur=1 THEN
CouleurGEG_Disjoncteur:=OR(intermediaire1,intermediaire2);
CouleurOnduleur_Disjoncteur:=CouleurGEG_Disjoncteur;
 ELSIF EtatCouplage2_Disjoncteur=0 THEN
        CouleurGEG_Disjoncteur:=intermediaire1;
        CouleurOnduleur_Disjoncteur:=intermediaire2;
END_IF;
CouleurCTA1_Disjoncteur:=CouleurGEG_Disjoncteur;
CouleurCTA2_Disjoncteur:=CouleurGEG_Disjoncteur;
CouleurCTA3_Disjoncteur:=CouleurGEG_Disjoncteur;
CouleurCTA4_Disjoncteur:=CouleurGEG_Disjoncteur;
CouleurASC1_Disjoncteur:=CouleurOnduleur_Disjoncteur;
CouleurASC2_Disjoncteur:=CouleurOnduleur_Disjoncteur;
CouleurASC3_Disjoncteur:=CouleurOnduleur_Disjoncteur;
CouleurPompe1_Disjoncteur:=CouleurOnduleur_Disjoncteur;
CouleurCouplage2_Disjoncteur:=CouleurOnduleur_Disjoncteur;

(*TGBT2*)
Couleur1Inv2_Selecteur:=AND(EtatTransfo3_Disjoncteur,CouleurTransfo3_Disjoncteur);
Couleur2Inv2_Selecteur:=AND(EtatGE5_Disjoncteur,CouleurGE5_Disjoncteur);
EtatInv2_Selecteur:=OR(etat_reseau1,etat_reseau2);
Couleur1Inv3_Selecteur:=AND(EtatTransfo4_Disjoncteur,CouleurTransfo4_Disjoncteur);
Couleur2Inv3_Selecteur:=AND(EtatGE6_Disjoncteur,CouleurGE6_Disjoncteur);
EtatInv3_Selecteur:=EtatInv2_Selecteur;
intermediaire1:=OR(Couleur1Inv2_Selecteur,Couleur2Inv2_Selecteur);
intermediaire2:=OR(Couleur1Inv3_Selecteur,Couleur2Inv3_Selecteur);
IF (intermediaire1=0)OR(intermediaire2=0) THEN
        EtatCouplage3_Disjoncteur:=1;
    ELSE
        EtatCouplage3_Disjoncteur:=0;
END_IF;
CouleurTDA1_Disjoncteur:=OR(intermediaire1,intermediaire2);
CouleurTDA2_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDA3_Disjoncteur:=CouleurTDA1_Disjoncteur;
```

```
CouleurTDA4_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDA5_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDA6_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTD_SSA_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTD_RDCA_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDB1_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDB2_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDB3_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDB4_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDB5_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTDB6_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTD_SSB_Disjoncteur:=CouleurTDA1_Disjoncteur;
CouleurTD_Ext_Disjoncteur:=CouleurTDB1_Disjoncteur;
CouleurCouplage3_Disjoncteur:=CouleurTDA1_Disjoncteur;

(*TGGE*)
FBI_27 (IN:=NOT(OR(etat_reseau1,etat_reseau2)), ZB:=T#3s,
        SW:=1,Q=>intermediaire1_tempo);
etat_GE1:=intermediaire1_tempo;
etat_GE2:=etat_GE1;
CouleurGE1_Disjoncteur:=OR(etat_GE1,etat_GE2);
CouleurGE4_Disjoncteur:=AND(EtatGE1_Disjoncteur,CouleurGE1_Disjoncteur);
CouleurGE5_Disjoncteur:=CouleurGE4_Disjoncteur;
CouleurGE6_Disjoncteur:=CouleurGE4_Disjoncteur;

(*TGCO*)
CouleurTGCO1_Disjoncteur:=Etat_Onduleur;
CouleurCCOA1_Disjoncteur:=AND(CouleurTGCO1_Disjoncteur,EtatTGCO1_Disjoncteur);
CouleurCCOA2_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOA3_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOA4_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOA5_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOA6_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOB1_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOB2_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOB3_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOB4_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOB5_Disjoncteur:=CouleurCCOA1_Disjoncteur;
CouleurCCOB6_Disjoncteur:=CouleurCCOA1_Disjoncteur;

(*TGBT2_Suite*)
Couleur1A_Disjoncteur:=AND(CouleurTDA1_Disjoncteur,EtatTDA1_Disjoncteur);
Couleur2A_Disjoncteur:=AND(CouleurTDA2_Disjoncteur,EtatTDA2_Disjoncteur);
Couleur3A_Disjoncteur:=AND(CouleurTDA3_Disjoncteur,EtatTDA3_Disjoncteur);
Couleur4A_Disjoncteur:=AND(CouleurTDA4_Disjoncteur,EtatTDA4_Disjoncteur);
Couleur5A_Disjoncteur:=AND(CouleurTDA5_Disjoncteur,EtatTDA5_Disjoncteur);
Couleur6A_Disjoncteur:=AND(CouleurTDA6_Disjoncteur,EtatTDA6_Disjoncteur);
Couleur1B_Disjoncteur:=AND(CouleurTDB1_Disjoncteur,EtatTDB1_Disjoncteur);
Couleur2B_Disjoncteur:=AND(CouleurTDB2_Disjoncteur,EtatTDB2_Disjoncteur);
Couleur3B_Disjoncteur:=AND(CouleurTDB3_Disjoncteur,EtatTDB3_Disjoncteur);
Couleur4B_Disjoncteur:=AND(CouleurTDB4_Disjoncteur,EtatTDB4_Disjoncteur);
Couleur5B_Disjoncteur:=AND(CouleurTDB5_Disjoncteur,EtatTDB5_Disjoncteur);
```

```
Couleur6B_Disjoncteur:=AND(CouleurTDB6_Disjoncteur,EtatTDB6_Disjoncteur);
CouleurSSA_Disjoncteur:=AND(CouleurTD_SSA_Disjoncteur,EtatTD_SSA_Disjoncteur);
CouleurRDCA_Disjoncteur:=AND(CouleurTD_RDCA_Disjoncteur,EtatTD_RDCA_Disjoncteur);
CouleurSSB_Disjoncteur:=AND(CouleurTD_SSB_Disjoncteur,EtatTD_SSB_Disjoncteur);
CouleurExt_Disjoncteur:=AND(CouleurTD_Ext_Disjoncteur,EtatTD_Ext_Disjoncteur);

(*TD1_AB*)
Couleur1A1A_Disjoncteur:=AND(Couleur1A_Disjoncteur,Etat1A_Disjoncteur);
Couleur1A2A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1A3A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1A4A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1A5A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1A6A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1A7A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1A8A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1A9A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1A10A_Disjoncteur:=Couleur1A1A_Disjoncteur;
Couleur1B1B_Disjoncteur:=AND(Couleur1B_Disjoncteur,Etat1B_Disjoncteur);
Couleur1B2B_Disjoncteur:=Couleur1B1B_Disjoncteur;
Couleur1B3B_Disjoncteur:=Couleur1B1B_Disjoncteur;
Couleur1B4B_Disjoncteur:=Couleur1B1B_Disjoncteur;
Couleur1B5B_Disjoncteur:=Couleur1B1B_Disjoncteur;
Couleur1B6B_Disjoncteur:=Couleur1B1B_Disjoncteur;
Couleur1B7B_Disjoncteur:=Couleur1B1B_Disjoncteur;
Courant1A1A:=AND(Etat1A1A_Disjoncteur,Couleur1A1A_Disjoncteur);
Courant1A2A:=AND(Etat1A2A_Disjoncteur,Couleur1A2A_Disjoncteur);
Courant1A3A:=AND(Etat1A3A_Disjoncteur,Couleur1A3A_Disjoncteur);
Courant1A4A:=AND(Etat1A4A_Disjoncteur,Couleur1A4A_Disjoncteur);
Courant1A5A:=AND(Etat1A5A_Disjoncteur,Couleur1A5A_Disjoncteur);
Courant1A6A:=AND(Etat1A6A_Disjoncteur,Couleur1A6A_Disjoncteur);
Courant1A7A:=AND(Etat1A7A_Disjoncteur,Couleur1A7A_Disjoncteur);
Courant1A8A:=AND(Etat1A8A_Disjoncteur,Couleur1A8A_Disjoncteur);
Courant1A9A:=AND(Etat1A9A_Disjoncteur,Couleur1A9A_Disjoncteur);
Courant1A10A:=AND(Etat1A10A_Disjoncteur,Couleur1A10A_Disjoncteur);
Courant1B1B:=AND(Etat1B1B_Disjoncteur,Couleur1B1B_Disjoncteur);
Courant1B2B:=AND(Etat1B2B_Disjoncteur,Couleur1B2B_Disjoncteur);
Courant1B3B:=AND(Etat1B3B_Disjoncteur,Couleur1B3B_Disjoncteur);
Courant1B4B:=AND(Etat1B4B_Disjoncteur,Couleur1B4B_Disjoncteur);
Courant1B5B:=AND(Etat1B5B_Disjoncteur,Couleur1B5B_Disjoncteur);
Courant1B6B:=AND(Etat1B6B_Disjoncteur,Couleur1B6B_Disjoncteur);
Courant1B7B:=AND(Etat1B7B_Disjoncteur,Couleur1B7B_Disjoncteur);
Etat1A_Groupe_Disjoncteur:=AND(Etat1A1A_Disjoncteur,Etat1A2A_Disjoncteur,Etat1A3A_Disjoncteur,Etat1
A4A_Disjoncteur,Etat1A5A_Disjoncteur,Etat1A6A_Disjoncteur,Etat1A7A_Disjoncteur,Etat1A8A_Disjoncteur,
Etat1A9A_Disjoncteur,Etat1A10A_Disjoncteur);
Etat1B_Groupe_Disjoncteur:=AND(Etat1B1B_Disjoncteur,Etat1B2B_Disjoncteur,Etat1B3B_Disjoncteur,Etat1
B4B_Disjoncteur,Etat1B5B_Disjoncteur,Etat1B6B_Disjoncteur,Etat1B7B_Disjoncteur);

(*eclairage exterieur*)
IF ModeCrep1=1 THEN
        Allumage_Zone1:=crepuscule;
END_IF;
IF ModeCrep2=1 THEN
```

```
        Allumage_Zone2:=crepuscule;
END_IF;
IF ModeCrep3=1 THEN
        Allumage_Zone3:=crepuscule;
END_IF;
IF ModeCrep4=1 THEN
        Allumage_Zone4:=crepuscule;
END_IF;
IF ModeCrep5=1 THEN
        Allumage_Zone5:=crepuscule;
END_IF;
IF ModeCrep6=1 THEN
        Allumage_Zone6:=crepuscule;
END_IF;
IF ModeCrep7=1 THEN
        Allumage_Zone7:=crepuscule;
END_IF;
IF ModeCrep8=1 THEN
        Allumage_Zone8:=crepuscule;
END_IF;

(*TD_Ext*)
CouleurZone1_Disjoncteur:=AND(CouleurExt_Disjoncteur,EtatExt_Disjoncteur);
CouleurZone2_Disjoncteur:=CouleurZone1_Disjoncteur;
CouleurZone3_Disjoncteur:=CouleurZone1_Disjoncteur;
CouleurZone4_Disjoncteur:=CouleurZone1_Disjoncteur;
CouleurZone5_Disjoncteur:=CouleurZone1_Disjoncteur;
CouleurZone6_Disjoncteur:=CouleurZone1_Disjoncteur;
CouleurZone7_Disjoncteur:=CouleurZone1_Disjoncteur;
CouleurZone8_Disjoncteur:=CouleurZone1_Disjoncteur;
CourantZone1:=AND(EtatZone1_Disjoncteur,CouleurZone1_Disjoncteur);
CourantZone2:=AND(EtatZone2_Disjoncteur,CouleurZone2_Disjoncteur);
CourantZone3:=AND(EtatZone3_Disjoncteur,CouleurZone3_Disjoncteur);
CourantZone4:=AND(EtatZone4_Disjoncteur,CouleurZone4_Disjoncteur);
CourantZone5:=AND(EtatZone5_Disjoncteur,CouleurZone5_Disjoncteur);
CourantZone6:=AND(EtatZone6_Disjoncteur,CouleurZone6_Disjoncteur);
CourantZone7:=AND(EtatZone7_Disjoncteur,CouleurZone7_Disjoncteur);
CourantZone8:=AND(EtatZone8_Disjoncteur,CouleurZone8_Disjoncteur);
EtatExt_Groupe_Disjoncteur:=AND(EtatZone1_Disjoncteur,EtatZone2_Disjoncteur,EtatZone3_Disjoncteur,Etat
Zone4_Disjoncteur,EtatZone5_Disjoncteur,EtatZone6_Disjoncteur,EtatZone7_Disjoncteur,EtatZone8_Disjoncte
ur);

(*TD2_AB*)
Couleur2A1A_Disjoncteur:=AND(Couleur2A_Disjoncteur,Etat2A_Disjoncteur);
Couleur2A2A_Disjoncteur:=Couleur2A1A_Disjoncteur;
Couleur2A3A_Disjoncteur:=Couleur2A1A_Disjoncteur;
Couleur2A4A_Disjoncteur:=Couleur2A1A_Disjoncteur;
Couleur2A5A_Disjoncteur:=Couleur2A1A_Disjoncteur;
Couleur2A6A_Disjoncteur:=Couleur2A1A_Disjoncteur;
Couleur2A7A_Disjoncteur:=Couleur2A1A_Disjoncteur;
Couleur2A8A_Disjoncteur:=Couleur2A1A_Disjoncteur;
Couleur2A9A_Disjoncteur:=Couleur2A1A_Disjoncteur;
```

```
Couleur2A10A_Disjoncteur:=Couleur2A1A_Disjoncteur;
Couleur2B1B_Disjoncteur:=AND(Couleur2B_Disjoncteur,Etat2B_Disjoncteur);
Couleur2B2B_Disjoncteur:=Couleur2B1B_Disjoncteur;
Couleur2B3B_Disjoncteur:=Couleur2B1B_Disjoncteur;
Couleur2B4B_Disjoncteur:=Couleur2B1B_Disjoncteur;
Couleur2B5B_Disjoncteur:=Couleur2B1B_Disjoncteur;
Couleur2B6B_Disjoncteur:=Couleur2B1B_Disjoncteur;
Couleur2B7B_Disjoncteur:=Couleur2B1B_Disjoncteur;
Courant2A1A:=AND(Etat2A1A_Disjoncteur,Couleur2A1A_Disjoncteur);
Courant2A2A:=AND(Etat2A2A_Disjoncteur,Couleur2A2A_Disjoncteur);
Courant2A3A:=AND(Etat2A3A_Disjoncteur,Couleur2A3A_Disjoncteur);
Courant2A4A:=AND(Etat2A4A_Disjoncteur,Couleur2A4A_Disjoncteur);
Courant2A5A:=AND(Etat2A5A_Disjoncteur,Couleur2A5A_Disjoncteur);
Courant2A6A:=AND(Etat2A6A_Disjoncteur,Couleur2A6A_Disjoncteur);
Courant2A7A:=AND(Etat2A7A_Disjoncteur,Couleur2A7A_Disjoncteur);
Courant2A8A:=AND(Etat2A8A_Disjoncteur,Couleur2A8A_Disjoncteur);
Courant2A9A:=AND(Etat2A9A_Disjoncteur,Couleur2A9A_Disjoncteur);
Courant2A10A:=AND(Etat2A10A_Disjoncteur,Couleur2A10A_Disjoncteur);
Courant2B1B:=AND(Etat2B1B_Disjoncteur,Couleur2B1B_Disjoncteur);
Courant2B2B:=AND(Etat2B2B_Disjoncteur,Couleur2B2B_Disjoncteur);
Courant2B3B:=AND(Etat2B3B_Disjoncteur,Couleur2B3B_Disjoncteur);
Courant2B4B:=AND(Etat2B4B_Disjoncteur,Couleur2B4B_Disjoncteur);
Courant2B5B:=AND(Etat2B5B_Disjoncteur,Couleur2B5B_Disjoncteur);
Courant2B6B:=AND(Etat2B6B_Disjoncteur,Couleur2B6B_Disjoncteur);
Courant2B7B:=AND(Etat2B7B_Disjoncteur,Couleur2B7B_Disjoncteur);
Etat2A_Groupe_Disjoncteur:=AND(Etat2A1A_Disjoncteur,Etat2A2A_Disjoncteur,Etat2A3A_Disjoncteur,Etat2
A4A_Disjoncteur,Etat2A5A_Disjoncteur,Etat2A6A_Disjoncteur,Etat2A7A_Disjoncteur,Etat2A8A_Disjoncteur,
Etat2A9A_Disjoncteur,Etat2A10A_Disjoncteur);
Etat2B_Groupe_Disjoncteur:=AND(Etat2B1B_Disjoncteur,Etat2B2B_Disjoncteur,Etat2B3B_Disjoncteur,Etat2
B4B_Disjoncteur,Etat2B5B_Disjoncteur,Etat2B6B_Disjoncteur,Etat2B7B_Disjoncteur);
(*TD3_AB*)
Couleur3A1A_Disjoncteur:=AND(Couleur3A_Disjoncteur,Etat3A_Disjoncteur);
Couleur3A2A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3A3A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3A4A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3A5A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3A6A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3A7A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3A8A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3A9A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3A10A_Disjoncteur:=Couleur3A1A_Disjoncteur;
Couleur3B1B_Disjoncteur:=AND(Couleur3B_Disjoncteur,Etat3B_Disjoncteur);
Couleur3B2B_Disjoncteur:=Couleur3B1B_Disjoncteur;
Couleur3B3B_Disjoncteur:=Couleur3B1B_Disjoncteur;
Couleur3B4B_Disjoncteur:=Couleur3B1B_Disjoncteur;
Couleur3B5B_Disjoncteur:=Couleur3B1B_Disjoncteur;
Couleur3B6B_Disjoncteur:=Couleur3B1B_Disjoncteur;
Couleur3B7B_Disjoncteur:=Couleur3B1B_Disjoncteur;
Courant3A1A:=AND(Etat3A1A_Disjoncteur,Couleur3A1A_Disjoncteur);
Courant3A2A:=AND(Etat3A2A_Disjoncteur,Couleur3A2A_Disjoncteur);
Courant3A3A:=AND(Etat3A3A_Disjoncteur,Couleur3A3A_Disjoncteur);
Courant3A4A:=AND(Etat3A4A_Disjoncteur,Couleur3A4A_Disjoncteur);
```

```
Courant3A5A:=AND(Etat3A5A_Disjoncteur,Couleur3A5A_Disjoncteur);
Courant3A6A:=AND(Etat3A6A_Disjoncteur,Couleur3A6A_Disjoncteur);
Courant3A7A:=AND(Etat3A7A_Disjoncteur,Couleur3A7A_Disjoncteur);
Courant3A8A:=AND(Etat3A8A_Disjoncteur,Couleur3A8A_Disjoncteur);
Courant3A9A:=AND(Etat3A9A_Disjoncteur,Couleur3A9A_Disjoncteur);
Courant3A10A:=AND(Etat3A10A_Disjoncteur,Couleur3A10A_Disjoncteur);
Courant3B1B:=AND(Etat3B1B_Disjoncteur,Couleur3B1B_Disjoncteur);
Courant3B2B:=AND(Etat3B2B_Disjoncteur,Couleur3B2B_Disjoncteur);
Courant3B3B:=AND(Etat3B3B_Disjoncteur,Couleur3B3B_Disjoncteur);
Courant3B4B:=AND(Etat3B4B_Disjoncteur,Couleur3B4B_Disjoncteur);
Courant3B5B:=AND(Etat3B5B_Disjoncteur,Couleur3B5B_Disjoncteur);
Courant3B6B:=AND(Etat3B6B_Disjoncteur,Couleur3B6B_Disjoncteur);
Courant3B7B:=AND(Etat3B7B_Disjoncteur,Couleur3B7B_Disjoncteur);
Etat3A_Groupe_Disjoncteur:=AND(Etat3A1A_Disjoncteur,Etat3A2A_Disjoncteur,Etat3A3A_Disjoncteur,Etat3
A4A_Disjoncteur,Etat3A5A_Disjoncteur,Etat3A6A_Disjoncteur,Etat3A7A_Disjoncteur,Etat3A8A_Disjoncteur,
Etat3A9A_Disjoncteur,Etat3A10A_Disjoncteur);
Etat3B_Groupe_Disjoncteur:=AND(Etat3B1B_Disjoncteur,Etat3B2B_Disjoncteur,Etat3B3B_Disjoncteur,Etat3
B4B_Disjoncteur,Etat3B5B_Disjoncteur,Etat3B6B_Disjoncteur,Etat3B7B_Disjoncteur);

(*TD4_AB*)
Couleur4A1A_Disjoncteur:=AND(Couleur4A_Disjoncteur,Etat4A_Disjoncteur);
Couleur4A2A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4A3A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4A4A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4A5A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4A6A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4A7A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4A8A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4A9A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4A10A_Disjoncteur:=Couleur4A1A_Disjoncteur;
Couleur4B1B_Disjoncteur:=AND(Couleur4B_Disjoncteur,Etat4B_Disjoncteur);
Couleur4B2B_Disjoncteur:=Couleur4B1B_Disjoncteur;
Couleur4B3B_Disjoncteur:=Couleur4B1B_Disjoncteur;
Couleur4B4B_Disjoncteur:=Couleur4B1B_Disjoncteur;
Couleur4B5B_Disjoncteur:=Couleur4B1B_Disjoncteur;
Couleur4B6B_Disjoncteur:=Couleur4B1B_Disjoncteur;
Couleur4B7B_Disjoncteur:=Couleur4B1B_Disjoncteur;
Courant4A1A:=AND(Etat4A1A_Disjoncteur,Couleur4A1A_Disjoncteur);
Courant4A2A:=AND(Etat4A2A_Disjoncteur,Couleur4A2A_Disjoncteur);
Courant4A3A:=AND(Etat4A3A_Disjoncteur,Couleur4A3A_Disjoncteur);
Courant4A4A:=AND(Etat4A4A_Disjoncteur,Couleur4A4A_Disjoncteur);
Courant4A5A:=AND(Etat4A5A_Disjoncteur,Couleur4A5A_Disjoncteur);
Courant4A6A:=AND(Etat4A6A_Disjoncteur,Couleur4A6A_Disjoncteur);
Courant4A7A:=AND(Etat4A7A_Disjoncteur,Couleur4A7A_Disjoncteur);
Courant4A8A:=AND(Etat4A8A_Disjoncteur,Couleur4A8A_Disjoncteur);
Courant4A9A:=AND(Etat4A9A_Disjoncteur,Couleur4A9A_Disjoncteur);
Courant4A10A:=AND(Etat4A10A_Disjoncteur,Couleur4A10A_Disjoncteur);
Courant4B1B:=AND(Etat4B1B_Disjoncteur,Couleur4B1B_Disjoncteur);
Courant4B2B:=AND(Etat4B2B_Disjoncteur,Couleur4B2B_Disjoncteur);
Courant4B3B:=AND(Etat4B3B_Disjoncteur,Couleur4B3B_Disjoncteur);
Courant4B4B:=AND(Etat4B4B_Disjoncteur,Couleur4B4B_Disjoncteur);
Courant4B5B:=AND(Etat4B5B_Disjoncteur,Couleur4B5B_Disjoncteur);
```

```
Courant4B6B:=AND(Etat4B6B_Disjoncteur,Couleur4B6B_Disjoncteur);
Courant4B7B:=AND(Etat4B7B_Disjoncteur,Couleur4B7B_Disjoncteur);
Etat4A_Groupe_Disjoncteur:=AND(Etat4A1A_Disjoncteur,Etat4A2A_Disjoncteur,Etat4A3A_Disjoncteur,Etat4A4A_Disjoncteur,Etat4A5A_Disjoncteur,Etat4A6A_Disjoncteur,Etat4A7A_Disjoncteur,Etat4A8A_Disjoncteur,Etat4A9A_Disjoncteur,Etat4A10A_Disjoncteur);
Etat4B_Groupe_Disjoncteur:=AND(Etat4B1B_Disjoncteur,Etat4B2B_Disjoncteur,Etat4B3B_Disjoncteur,Etat4B4B_Disjoncteur,Etat4B5B_Disjoncteur,Etat4B6B_Disjoncteur,Etat4B7B_Disjoncteur);

(*TD5_AB*)
Couleur5A1A_Disjoncteur:=AND(Couleur5A_Disjoncteur,Etat5A_Disjoncteur);
Couleur5A2A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5A3A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5A4A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5A5A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5A6A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5A7A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5A8A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5A9A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5A10A_Disjoncteur:=Couleur5A1A_Disjoncteur;
Couleur5B1B_Disjoncteur:=AND(Couleur5B_Disjoncteur,Etat5B_Disjoncteur);
Couleur5B2B_Disjoncteur:=Couleur5B1B_Disjoncteur;
Couleur5B3B_Disjoncteur:=Couleur5B1B_Disjoncteur;
Couleur5B4B_Disjoncteur:=Couleur5B1B_Disjoncteur;
Couleur5B5B_Disjoncteur:=Couleur5B1B_Disjoncteur;
Couleur5B6B_Disjoncteur:=Couleur5B1B_Disjoncteur;
Couleur5B7B_Disjoncteur:=Couleur5B1B_Disjoncteur;
Courant5A1A:=AND(Etat5A1A_Disjoncteur,Couleur5A1A_Disjoncteur);
Courant5A2A:=AND(Etat5A2A_Disjoncteur,Couleur5A2A_Disjoncteur);
Courant5A3A:=AND(Etat5A3A_Disjoncteur,Couleur5A3A_Disjoncteur);
Courant5A4A:=AND(Etat5A4A_Disjoncteur,Couleur5A4A_Disjoncteur);
Courant5A5A:=AND(Etat5A5A_Disjoncteur,Couleur5A5A_Disjoncteur);
Courant5A6A:=AND(Etat5A6A_Disjoncteur,Couleur5A6A_Disjoncteur);
Courant5A7A:=AND(Etat5A7A_Disjoncteur,Couleur5A7A_Disjoncteur);
Courant5A8A:=AND(Etat5A8A_Disjoncteur,Couleur5A8A_Disjoncteur);
Courant5A9A:=AND(Etat5A9A_Disjoncteur,Couleur5A9A_Disjoncteur);
Courant5A10A:=AND(Etat5A10A_Disjoncteur,Couleur5A10A_Disjoncteur);
Courant5B1B:=AND(Etat5B1B_Disjoncteur,Couleur5B1B_Disjoncteur);
Courant5B2B:=AND(Etat5B2B_Disjoncteur,Couleur5B2B_Disjoncteur);
Courant5B3B:=AND(Etat5B3B_Disjoncteur,Couleur5B3B_Disjoncteur);
Courant5B4B:=AND(Etat5B4B_Disjoncteur,Couleur5B4B_Disjoncteur);
Courant5B5B:=AND(Etat5B5B_Disjoncteur,Couleur5B5B_Disjoncteur);
Courant5B6B:=AND(Etat5B6B_Disjoncteur,Couleur5B6B_Disjoncteur);
Courant5B7B:=AND(Etat5B7B_Disjoncteur,Couleur5B7B_Disjoncteur);
Etat5A_Groupe_Disjoncteur:=AND(Etat5A1A_Disjoncteur,Etat5A2A_Disjoncteur,Etat5A3A_Disjoncteur,Etat5A4A_Disjoncteur,Etat5A5A_Disjoncteur,Etat5A6A_Disjoncteur,Etat5A7A_Disjoncteur,Etat5A8A_Disjoncteur,Etat5A9A_Disjoncteur,Etat5A10A_Disjoncteur);
Etat5B_Groupe_Disjoncteur:=AND(Etat5B1B_Disjoncteur,Etat5B2B_Disjoncteur,Etat5B3B_Disjoncteur,Etat5B4B_Disjoncteur,Etat5B5B_Disjoncteur,Etat5B6B_Disjoncteur,Etat5B7B_Disjoncteur);

(*TD6_AB*)
Couleur6A1A_Disjoncteur:=AND(Couleur6A_Disjoncteur,Etat6A_Disjoncteur);
```

```
Couleur6A2A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6A3A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6A4A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6A5A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6A6A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6A7A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6A8A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6A9A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6A10A_Disjoncteur:=Couleur6A1A_Disjoncteur;
Couleur6B1B_Disjoncteur:=AND(Couleur6B_Disjoncteur,Etat6B_Disjoncteur);
Couleur6B2B_Disjoncteur:=Couleur6B1B_Disjoncteur;
Couleur6B3B_Disjoncteur:=Couleur6B1B_Disjoncteur;
Couleur6B4B_Disjoncteur:=Couleur6B1B_Disjoncteur;
Couleur6B5B_Disjoncteur:=Couleur6B1B_Disjoncteur;
Couleur6B6B_Disjoncteur:=Couleur6B1B_Disjoncteur;
Couleur6B7B_Disjoncteur:=Couleur6B1B_Disjoncteur;
Courant6A1A:=AND(Etat6A1A_Disjoncteur,Couleur6A1A_Disjoncteur);
Courant6A2A:=AND(Etat6A2A_Disjoncteur,Couleur6A2A_Disjoncteur);
Courant6A3A:=AND(Etat6A3A_Disjoncteur,Couleur6A3A_Disjoncteur);
Courant6A4A:=AND(Etat6A4A_Disjoncteur,Couleur6A4A_Disjoncteur);
Courant6A5A:=AND(Etat6A5A_Disjoncteur,Couleur6A5A_Disjoncteur);
Courant6A6A:=AND(Etat6A6A_Disjoncteur,Couleur6A6A_Disjoncteur);
Courant6A7A:=AND(Etat6A7A_Disjoncteur,Couleur6A7A_Disjoncteur);
Courant6A8A:=AND(Etat6A8A_Disjoncteur,Couleur6A8A_Disjoncteur);
Courant6A9A:=AND(Etat6A9A_Disjoncteur,Couleur6A9A_Disjoncteur);
Courant6A10A:=AND(Etat6A10A_Disjoncteur,Couleur6A10A_Disjoncteur);
Courant6B1B:=AND(Etat6B1B_Disjoncteur,Couleur6B1B_Disjoncteur);
Courant6B2B:=AND(Etat6B2B_Disjoncteur,Couleur6B2B_Disjoncteur);
Courant6B3B:=AND(Etat6B3B_Disjoncteur,Couleur6B3B_Disjoncteur);
Courant6B4B:=AND(Etat6B4B_Disjoncteur,Couleur6B4B_Disjoncteur);
Courant6B5B:=AND(Etat6B5B_Disjoncteur,Couleur6B5B_Disjoncteur);
Courant6B6B:=AND(Etat6B6B_Disjoncteur,Couleur6B6B_Disjoncteur);
Courant6B7B:=AND(Etat6B7B_Disjoncteur,Couleur6B7B_Disjoncteur);
Etat6A_Groupe_Disjoncteur:=AND(Etat6A1A_Disjoncteur,Etat6A2A_Disjoncteur,Etat6A3A_Disjoncteur,Etat6
A4A_Disjoncteur,Etat6A5A_Disjoncteur,Etat6A6A_Disjoncteur,Etat6A7A_Disjoncteur,Etat6A8A_Disjoncteur,
Etat6A9A_Disjoncteur,Etat6A10A_Disjoncteur);
Etat6B_Groupe_Disjoncteur:=AND(Etat6B1B_Disjoncteur,Etat6B2B_Disjoncteur,Etat6B3B_Disjoncteur,Etat6
B4B_Disjoncteur,Etat6B5B_Disjoncteur,Etat6B6B_Disjoncteur,Etat6B7B_Disjoncteur);

(*TDSS_AB*)
CouleurSSA1A_Disjoncteur:=AND(CouleurSSA_Disjoncteur,EtatSSA_Disjoncteur);
CouleurSSA2A_Disjoncteur:=CouleurSSA1A_Disjoncteur;
CouleurSSA3A_Disjoncteur:=CouleurSSA1A_Disjoncteur;
CouleurSSA4A_Disjoncteur:=CouleurSSA1A_Disjoncteur;
CouleurSSA5A_Disjoncteur:=CouleurSSA1A_Disjoncteur;
CouleurSSA6A_Disjoncteur:=CouleurSSA1A_Disjoncteur;
CouleurSSB1B_Disjoncteur:=AND(CouleurSSB_Disjoncteur,EtatSSB_Disjoncteur);
CouleurSSB2B_Disjoncteur:=CouleurSSB1B_Disjoncteur;
CouleurSSB3B_Disjoncteur:=CouleurSSB1B_Disjoncteur;
CouleurSSB4B_Disjoncteur:=CouleurSSB1B_Disjoncteur;
CourantSSA1A:=AND(EtatSSA1A_Disjoncteur,CouleurSSA1A_Disjoncteur);
CourantSSA2A:=AND(EtatSSA2A_Disjoncteur,CouleurSSA2A_Disjoncteur);
```

```
CourantSSA3A:=AND(EtatSSA3A_Disjoncteur,CouleurSSA3A_Disjoncteur);
CourantSSA4A:=AND(EtatSSA4A_Disjoncteur,CouleurSSA4A_Disjoncteur);
CourantSSA5A:=AND(EtatSSA5A_Disjoncteur,CouleurSSA5A_Disjoncteur);
CourantSSA6A:=AND(EtatSSA6A_Disjoncteur,CouleurSSA6A_Disjoncteur);
CourantSSB1B:=AND(EtatSSB1B_Disjoncteur,CouleurSSB1B_Disjoncteur);
CourantSSB2B:=AND(EtatSSB2B_Disjoncteur,CouleurSSB2B_Disjoncteur);
CourantSSB3B:=AND(EtatSSB3B_Disjoncteur,CouleurSSB3B_Disjoncteur);
CourantSSB4B:=AND(EtatSSB4B_Disjoncteur,CouleurSSB4B_Disjoncteur);
EtatSSA_Groupe_Disjoncteur:=AND(EtatSSA1A_Disjoncteur,EtatSSA2A_Disjoncteur,EtatSSA3A_Disjoncteu
r,EtatSSA4A_Disjoncteur,EtatSSA5A_Disjoncteur,EtatSSA6A_Disjoncteur);
EtatSSB_Groupe_Disjoncteur:=AND(EtatSSB1B_Disjoncteur,EtatSSB2B_Disjoncteur,EtatSSB3B_Disjoncteur,
EtatSSB4B_Disjoncteur);

(*TD_RDC_A*)
CouleurRDCA1A_Disjoncteur:=AND(CouleurRDCA_Disjoncteur,EtatRDCA_Disjoncteur);
CouleurRDCA2A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA3A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA4A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA5A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA6A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA7A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA8A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA9A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA10A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA11A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA12A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA13A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CouleurRDCA14A_Disjoncteur:=CouleurRDCA1A_Disjoncteur;
CourantRDCA1A:=AND(EtatRDCA1A_Disjoncteur,CouleurRDCA1A_Disjoncteur);
CourantRDCA2A:=AND(EtatRDCA2A_Disjoncteur,CouleurRDCA2A_Disjoncteur);
CourantRDCA3A:=AND(EtatRDCA3A_Disjoncteur,CouleurRDCA3A_Disjoncteur);
CourantRDCA4A:=AND(EtatRDCA4A_Disjoncteur,CouleurRDCA4A_Disjoncteur);
CourantRDCA5A:=AND(EtatRDCA5A_Disjoncteur,CouleurRDCA5A_Disjoncteur);
CourantRDCA6A:=AND(EtatRDCA6A_Disjoncteur,CouleurRDCA6A_Disjoncteur);
CourantRDCA7A:=AND(EtatRDCA7A_Disjoncteur,CouleurRDCA7A_Disjoncteur);
CourantRDCA8A:=AND(EtatRDCA8A_Disjoncteur,CouleurRDCA8A_Disjoncteur);
CourantRDCA9A:=AND(EtatRDCA9A_Disjoncteur,CouleurRDCA9A_Disjoncteur);
CourantRDCA10A:=AND(EtatRDCA10A_Disjoncteur,CouleurRDCA10A_Disjoncteur);
CourantRDCA11A:=AND(EtatRDCA11A_Disjoncteur,CouleurRDCA11A_Disjoncteur);
CourantRDCA12A:=AND(EtatRDCA12A_Disjoncteur,CouleurRDCA12A_Disjoncteur);
CourantRDCA13A:=AND(EtatRDCA13A_Disjoncteur,CouleurRDCA13A_Disjoncteur);
CourantRDCA14A:=AND(EtatRDCA14A_Disjoncteur,CouleurRDCA14A_Disjoncteur);
EtatRDCA_Groupe_Disjoncteur:=AND(EtatRDCA1A_Disjoncteur,EtatRDCA2A_Disjoncteur,EtatRDCA3A_D
isjoncteur,EtatRDCA4A_Disjoncteur,EtatRDCA5A_Disjoncteur,EtatRDCA6A_Disjoncteur,EtatRDCA7A_Disj
oncteur,EtatRDCA8A_Disjoncteur,EtatRDCA9A_Disjoncteur,EtatRDCA10A_Disjoncteur,EtatRDCA10A_Disj
oncteur,
EtatRDCA11A_Disjoncteur,EtatRDCA12A_Disjoncteur,EtatRDCA13A_Disjoncteur,EtatRDCA14A_Disjoncte
ur);
```

1.3 Programmation de la DFB pour la Régulation de la température d'air soufflé de la CTA (Centrale de Traitement d'Air)

```
IF Activation=0 THEN
        terme_Integral:=0.0;
        PC:=0.0;
        Schaud:=0.0;
        Sfroid:=0.0;

  ELSE
  (*régulation PI*)
        e:=Car-Tar;
        terme_Integral:=terme_Integral+e;
        PC_calcul:=Gain*(e+(pas/Ti)*terme_Integral);
        IF PC_calcul<=PC_min THEN
                PC:=PC_min;
      ELSIF PC_calcul>=PC_max THEN
        PC:=PC_max;
       ELSE
        PC:=PC_calcul;
    END_IF;

   (*régulation Split Range*)
        Schaud_calcul:=(-100.0/(PC-1.0))*Tas+100.0;
        Sfroid_calcul:=(100.0/(PC+1.0))*Tas-100.0;
        IF Schaud_calcul<=0.0 THEN
                Schaud:=0.0;
        ELSIF Schaud_calcul>=100.0 THEN
        Schaud:=100.0;
       ELSE
        Schaud:=Schaud_calcul;
    END_IF;
        IF Sfroid_calcul<=0.0 THEN
                Sfroid:=0.0;

      ELSIF Sfroid_calcul>=100.0 THEN
        Sfroid:=100.0;
       ELSE
        Sfroid:=Sfroid_calcul;
    END_IF;
END_IF;
```

1.4 Programme de Gestion de la CTA

1.4.1 Partie programmée en ST

```
(*CTA1*)
IF Allumage_CTA1=1 THEN
        MarcheVentilateur_CTA1:=1;
        IF (AlarmeVentilo_CTA1=0)AND(AlarmeBatElec_CTA1=0) THEN
        Regulation_CTA1:=1;
        END_IF;
```

```
END_IF;

IF (Allumage_CTA1=0)OR(AlarmeVentilo_CTA1=1)OR(AlarmeBatElec_CTA1=1) THEN
        Regulation_CTA1:=0;
        FBI_19 (IN:=NOT(Allumage_CTA1), ZB:=T#5s,
        SW:=1,Q=>tempo_CTA1);
        IF (tempo_CTA1=1) THEN
        MarcheVentilateur_CTA1:=0;
        tempo_CTA1:=0;
        END_IF;
END_IF;

(*CTA2*)
IF Allumage_CTA2=1 THEN
        MarcheVentilateur_CTA2:=1;
        IF (AlarmeVentilo_CTA2=0)AND(AlarmeBatElec_CTA2=0) THEN
        Regulation_CTA2:=1;
        END_IF;
END_IF;

IF (Allumage_CTA2=0)OR(AlarmeVentilo_CTA2=1)OR(AlarmeBatElec_CTA2=1) THEN
        Regulation_CTA2:=0;
        FBI_19 (IN:=NOT(Allumage_CTA2), ZB:=T#5s,
        SW:=1,Q=>tempo_CTA2);
        IF (tempo_CTA2=1) THEN
        MarcheVentilateur_CTA2:=0;
        tempo_CTA2:=0;
        END_IF;
END_IF;

(*CTA3*)
IF Allumage_CTA3=1 THEN
        MarcheVentilateur_CTA3:=1;
        IF (AlarmeVentilo_CTA3=0)AND(AlarmeBatElec_CTA3=0) THEN
        Regulation_CTA3:=1;
        END_IF;
END_IF;

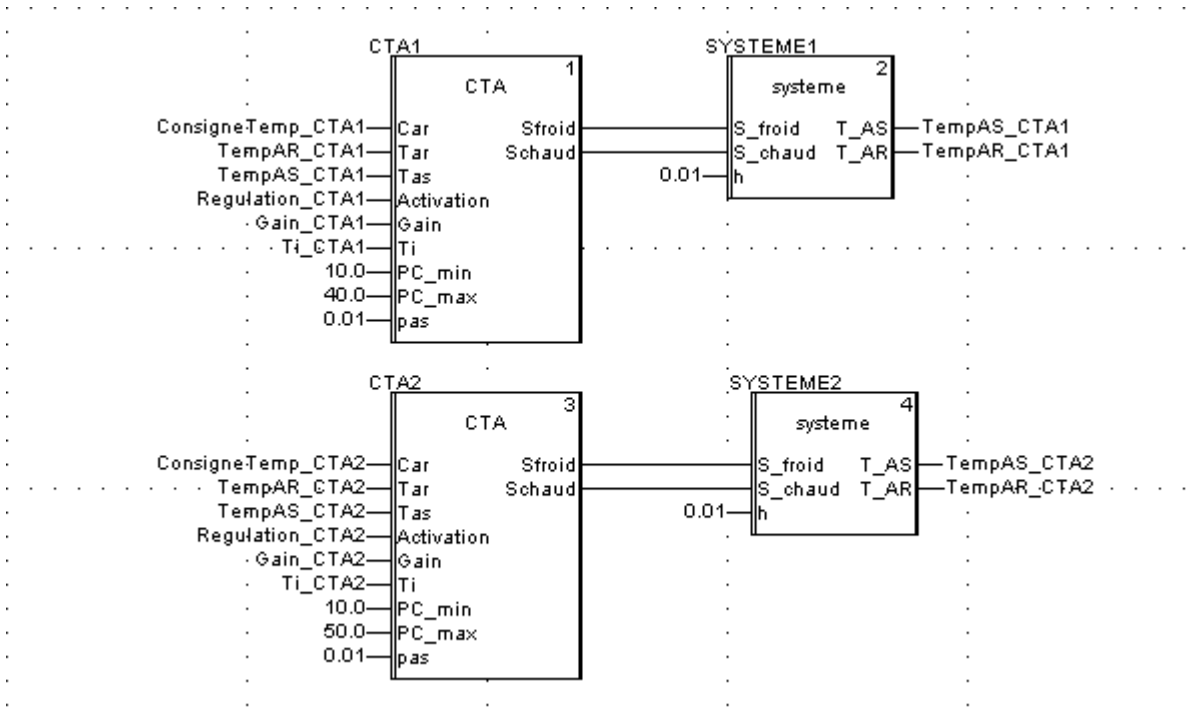
IF (Allumage_CTA3=0)OR(AlarmeVentilo_CTA3=1)OR(AlarmeBatElec_CTA3=1) THEN
        Regulation_CTA3:=0;
        FBI_19 (IN:=NOT(Allumage_CTA3), ZB:=T#5s,
        SW:=1,Q=>tempo_CTA3);
        IF (tempo_CTA3=1) THEN
        MarcheVentilateur_CTA3:=0;
        tempo_CTA3:=0;
        END_IF;
END_IF;

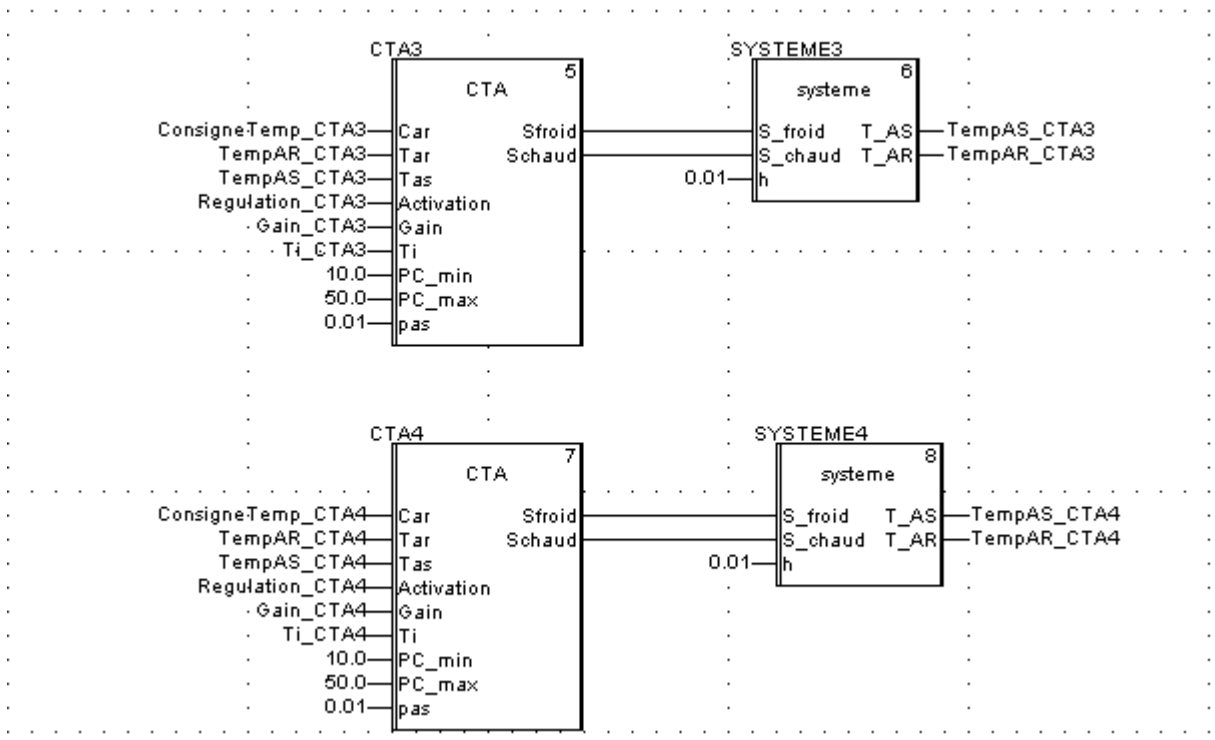
(*CTA4*)
IF Allumage_CTA4=1 THEN
        MarcheVentilateur_CTA4:=1;
        IF (AlarmeVentilo_CTA4=0)AND(AlarmeBatElec_CTA4=0) THEN
```

```
        Regulation_CTA4:=1;
        END_IF;
END_IF;

IF (Allumage_CTA4=0)OR(AlarmeVentilo_CTA4=1)OR(AlarmeBatElec_CTA4=1) THEN
        Regulation_CTA4:=0;
        FBI_19 (IN:=NOT(Allumage_CTA4), ZB:=T#5s,
        SW:=1,Q=>tempo_CTA4);
        IF (tempo_CTA4=1) THEN
        MarcheVentilateur_CTA4:=0;
        tempo_CTA4:=0;
        END_IF;
END_IF;
```

1.4.2 Partie programmée en FBD (partie simulation)





2. Programmes en Cicode (sous Vijeo Citect)

2.1 Programme de Gestion horaire (d'Allumage/Extinction des luminaires) des étages

Dans ce qui suit nous allons présenter le programme écrit en Cicode (langage de programmation sous Vijeo Citect) de la gestion horaire pour l'allumage et l'extinction des luminaires de la zone 1A du 1er étage.

Pour les autres étages il faudra apporter les modifications appropriés (modification du nom des variables)

Les programme est le suivant :

```
INT FUNCTION
ValiderChangementEt1A1(INT HeureAH, INT HeureAMn, INT HeureEH, INT
HeureEMn)

WHILE TRUE DO

    HeureActuelleH = TimeHour(TimeCurrent);
    HeureActuelleMn = TimeMin(TimeCurrent);
    IF HeureAH=HeureActuelleH
        THEN IF HeureAMn=HeureActuelleMn
                THEN IF Mode_Etage1_1A=1
                    THEN  Allumage_Etage1_1A=1;
                    END
            END
```

```
    END

    IF HeureEH=HeureActuelleH
        THEN IF HeureActuelleMn=HeureEMn
            THEN IF Mode_Etage1_1A=1
                THEN  Allumage_Etage1_1A= 0;
                END
            END
    END
    Sleep(10);

END
RETURN(0);
END
```

2.1 Programme de Gestion horaire (d'Allumage/Extinction des luminaires) huit zones d'éclairage extérieur

```
INT FUNCTION
ValiderChangementExt(INT HeureAH, INT HeureAMn, INT HeureEH, INT HeureEMn)

WHILE TRUE DO

        HeureActuelleH = TimeHour(TimeCurrent);
        HeureActuelleMn = TimeMin(TimeCurrent);
        IF HeureAH=HeureActuelleH
            THEN IF HeureAMn=HeureActuelleMn
                THEN IF (ModeZone1=1)AND(ModeCrep1=0)
                        THEN  Allumage_Zone1=1;
                    END
            END
        END


        IF HeureEH=HeureActuelleH
            THEN IF HeureActuelleMn=HeureEMn
                THEN IF (ModeZone1=1)AND(ModeCrep1=0)
                        THEN  Allumage_Zone1=0;
                    END
                END
        END

    IF HeureAH=HeureActuelleH
            THEN IF HeureAMn=HeureActuelleMn
                THEN IF (ModeZone2=1)AND(ModeCrep2=0)
                        THEN  Allumage_Zone2=1;
                    END
            END
        END


        IF HeureEH=HeureActuelleH
            THEN IF HeureActuelleMn=HeureEMn
                THEN IF (ModeZone2=1)AND(ModeCrep2=0)
                        THEN  Allumage_Zone2=0;
                    END
                END
```

```
        END

    IF HeureAH=HeureActuelleH
            THEN IF HeureAMn=HeureActuelleMn
                THEN IF (ModeZone3=1)AND(ModeCrep3=0)
                        THEN  Allumage_Zone3=1;
                    END
            END
        END


        IF HeureEH=HeureActuelleH
            THEN IF HeureActuelleMn=HeureEMn
                THEN IF (ModeZone3=1)AND(ModeCrep3=0)
                        THEN  Allumage_Zone3=0;
                    END
                END
        END

    IF HeureAH=HeureActuelleH
            THEN IF HeureAMn=HeureActuelleMn
                THEN IF (ModeZone4=1)AND(ModeCrep4=0)
                        THEN  Allumage_Zone4=1;
                    END
            END
        END


        IF HeureEH=HeureActuelleH
            THEN IF HeureActuelleMn=HeureEMn
                THEN IF (ModeZone4=1)AND(ModeCrep4=0)
                        THEN  Allumage_Zone4=0;
                    END
                END
        END

    IF HeureAH=HeureActuelleH
            THEN IF HeureAMn=HeureActuelleMn
                THEN IF (ModeZone5=1)AND(ModeCrep5=0)
                        THEN  Allumage_Zone5=1;
                    END
            END
        END


        IF HeureEH=HeureActuelleH
            THEN IF HeureActuelleMn=HeureEMn
                THEN IF (ModeZone5=1)AND(ModeCrep5=0)
                        THEN  Allumage_Zone5=0;
                    END
                END
        END

    IF HeureAH=HeureActuelleH
            THEN IF HeureAMn=HeureActuelleMn
                THEN IF (ModeZone6=1)AND(ModeCrep6=0)
                        THEN  Allumage_Zone6=1;
                    END
            END
        END
```

```
        IF HeureEH=HeureActuelleH
            THEN IF HeureActuelleMn=HeureEMn
                THEN IF (ModeZone6=1)AND(ModeCrep6=0)
                        THEN  Allumage_Zone6=0;
                    END
                END
        END

    IF HeureAH=HeureActuelleH
            THEN IF HeureAMn=HeureActuelleMn
                THEN IF (ModeZone7=1)AND(ModeCrep7=0)
                        THEN  Allumage_Zone7=1;
                    END
            END
        END


        IF HeureEH=HeureActuelleH
            THEN IF HeureActuelleMn=HeureEMn
                THEN IF (ModeZone7=1)AND(ModeCrep7=0)
                        THEN  Allumage_Zone7=0;
                    END
                END
        END

    IF HeureAH=HeureActuelleH
            THEN IF HeureAMn=HeureActuelleMn
                THEN IF (ModeZone8=1)AND(ModeCrep8=0)
                        THEN  Allumage_Zone8=1;
                    END
            END
        END


        IF HeureEH=HeureActuelleH
            THEN IF HeureActuelleMn=HeureEMn
                THEN IF (ModeZone8=1)AND(ModeCrep8=0)
                        THEN  Allumage_Zone8=0;
                    END
                END
        END


    Sleep(10);

END
RETURN(0);
END
```

2.1 Programme de Gestion horaire (Marche/Arrêt) de la CTA

Dans ce qui suit nous allons présenter le programme écrit en Cicode de la gestion horaire pour la Marche et l'arrêt de la première CTA (CTA1)

Pour les autres CTA, il faudra apporter les modifications appropriés (modification du nom des variables)

Les programme est le suivant :

```
INT FUNCTION
ValiderChangementCTA1(INT HeureAH, INT HeureAMn, INT HeureEH, INT HeureEMn)

WHILE TRUE DO

    HeureActuelleH = TimeHour(TimeCurrent);
    HeureActuelleMn = TimeMin(TimeCurrent);
    IF HeureAH=HeureActuelleH
        THEN IF HeureAMn=HeureActuelleMn
                THEN IF Mode_CTA1=1
                    THEN  Allumage_CTA1=1;
                    END
            END
    END

    IF HeureEH=HeureActuelleH
        THEN IF HeureActuelleMn=HeureEMn
            THEN IF Mode_CTA1=1
                THEN  Allumage_CTA1= 0;
                END
            END
    END
    Sleep(10);

END
RETURN(0);
END
```